Segmentation of Lumbar Vertebrae Slices from CT Images

Hugo Hutt, Richard Everson and Judith Meakin

Abstract We describe a fully automated approach to vertebrae segmentation from CT images which operates on superpixels. The method is based on a conditional random field model incorporating constraints learned from labelled superpixel features. The method is shown to provide consistently accurate segmentations of different vertebrae from a variety of subjects.

1 Introduction

Automatic segmentation of vertebrae from CT images is a challenging problem due to the complex and varied shape of the vertebrae, in addition to the various artefacts which may result from the acquisition process. However, segmenting the vertebrae by hand is a difficult and time consuming process. Automated segmentation is therefore desired to obtain reliable and accurate segmentations on any large scale.

Much of the previous work in this area has concentrated on sagittal views to provide segmentation of many vertebrae and intervening discs. With this view the pedicles and posterior elements of the vertebrae are frequently not visible, so segmentation has focussed on the vertebral bodies and employed tools such as statistical shape models and appearance models combined with probabilistic graphical models; e.g., [7, 11]. Huang et al. [8] have recently described a level set method for vertebrae segmentation using transverse (axial) CT slices.

In this paper, we describe a fully automated segmentation method that effectively segments the whole vertebra structure (including pedicles and posterior elements)

H. Hutt (✉) · R. Everson · J. Meakin
University of Exeter, Exeter, UK
e-mail: hwh202@exeter.ac.uk

R. Everson
e-mail: R.M.Everson@exeter.ac.uk

J. Meakin
e-mail: J.R.Meakin@exeter.ac.uk

J. Yao et al. (eds.), *Recent Advances in Computational Methods and Clinical Applications for Spine Imaging*, Lecture Notes in Computational Vision and Biomechanics 20, DOI 10.1007/978-3-319-14148-0_6

from transverse (axial) views. Our method for CT images is adapted from a method developed for MR images, described in [9]. At heart, the method uses a conditional random field (CRF) model on superpixels. Operating on superpixels reduces computational complexity and enables more descriptive features to be extracted to characterise the vertebra (foreground) and non-vertebra (background) classes, while the CRF relates the underlying class labels of the superpixels to the observed features and promotes coherence. We use supervised learning to train a classifier on labelled superpixel features and obtain probability estimates expressing the likelihood of belonging to either the vertebra or background class. Distance metric learning [17] is also used to find an appropriate dissimilarity measure between superpixel pairs. The probability estimates and learned distance metric are incorporated into the CRF model in the form of first- and second-order clique potentials of the CRF energy function. This formulation enables minimisation of the energy function to be carried out efficiently using graph cuts [3].

We evaluate the performance of the method on CT data from a range of subjects collected for the Computational Methods and Clinical Applications for Spine Imaging (CSI 2014) segmentation competition. We show that consistently accurate segmentations can be obtained for each of the different lumbar vertebrae.

2 Segmentation Model

Our method is based on a conditional random field (CRF) [2] model which operates on the superpixels of an image; we denote the set of superpixels by $\mathcal{S}$. The energy function of the CRF defines a posterior probability distribution $P(\mathbf{x} \mid \mathbf{y})$ for a set of class labels $\mathbf{x}$ for the superpixels, given a set of features $\mathbf{y}$ describing the superpixels. The energy function can be written as a sum of first- and second-order potential functions in the form

$$E(\mathbf{x}, \mathbf{y}) = \sum_{i \in \mathcal{S}} \underbrace{\psi(\mathbf{y}_i \mid x_i)}_{\text{Data term}} + \lambda \sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{N}_i} \underbrace{\phi(\mathbf{y}_i, \mathbf{y}_j \mid x_i, x_j)}_{\text{Smoothness term}} \tag{1}$$

where $\mathcal{N}_i$ is the set of neighbours of superpixel i. The constant λ controls the relative importance of the data and smoothness terms. The CRF formulation enables maximum a posteriori (MAP) inference of the labels $\mathbf{x}$ to be carried out efficiently using graph cuts. We use the min-cut/max-flow algorithm of [4] to find the optimal solution.

We define the potential functions of (1) by using supervised learning on labelled superpixel features and deriving constraints using the resulting trained models. Sections 3 and 4 describe the superpixel features used to learn the constraints and how they are incorporated into the CRF potential functions.

3 Superpixels

We use the Simple Linear Iterative Clustering (SLIC) [1, 16] algorithm to partition the image into superpixels. As shown in Fig. 1, boundaries of superpixels tend to coincide with boundaries of anatomical objects, enabling an accurate pixel-level segmentation to be recovered from the classified superpixels. The primary advantages of using superpixels are twofold: firstly, as the number of nodes in the graph decreases significantly from a pixel-level graph, there is a corresponding reduction in computational complexity. Secondly, multiple features can be extracted from the superpixel regions which can help to discriminate between the classes more effectively.

We aim to characterise the superpixels by extracting multiple features from them that incorporate information about intensity, texture, location and edge response. As described in the next section, these features are used to discriminate between the vertebra and background superpixels by learning a classifier and distance metric on a set of ground truth images. We emphasise that this training occurs only once, after which the trained models can be used in the CRF potential functions for any further images.

The superpixel features are summarised in Table 1. The feature vector for a superpixel i is a concatenation of the individual features:

$$\mathbf{y}_i = [\mathbf{y}_i^T, \mathbf{y}_i^L, \mathbf{y}_i^E]^\top. \tag{2}$$

We exhaustively tested different subsets of the features, but found that the best performance was obtained by combining all features. The features were chosen in part

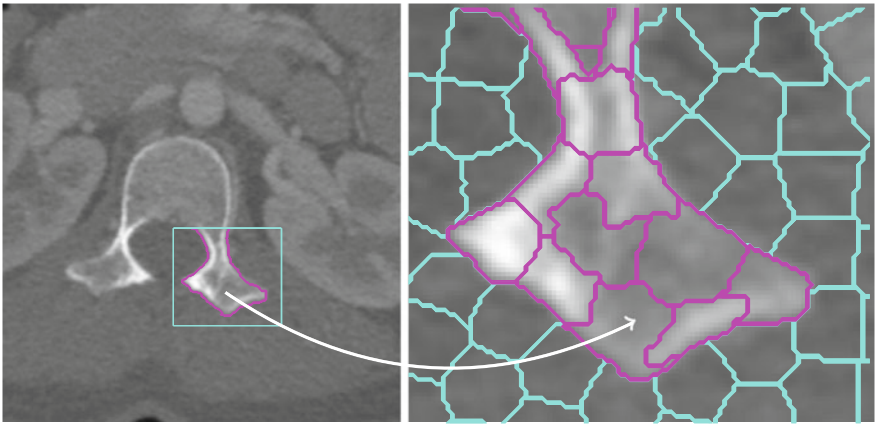

Fig. 1 The *left* figure shows a CT slice with ground truth contour (*magenta*) for a section of the vertebra. The *right* figure shows boundaries for superpixels assigned to the vertebra class (*magenta*) and background class (*cyan*). The superpixels preserve the boundary detail of the vertebrae (color figure online)

Table 1 Superpixel features (p_n denotes the nth percentile)

Feature	Description	Dimension
$\mathbf{y}_i^{T_1}$	Concatenation of intensity histogram from superpixel i and average histogram from neighbours $\mathcal{N}_i$	20
$\mathbf{y}_i^{T_2}$	SIFT descriptor calculated at the centroid of superpixel i	128
$\mathbf{y}_i^{L_1}$	Mean, p_{10} and p_{90} of the row and column pixel coordinates in the superpixel, centred on the matched contour region	6
$\mathbf{y}_i^{L_2}$	Mean, p_{10} and p_{90} of the matched contour distance transform gradient in the superpixel, in both the horizontal and vertical direction	6
$\mathbf{y}_i^{E_1}$	Mean, p_{10} and p_{90} of the LoG response within the superpixel, taken over 4 scales	12
$\mathbf{y}_i^{E_2}$	Mean, p_{10} and p_{90} of the structure tensor eigenvalues of the superpixel, taken over 4 scales	24

for their generality and as a consequence are directly applicable to different imaging modalities such as MRI [9].

The first set of features $\mathbf{y}_i^T$ characterise the intensity and textural properties of the superpixels. They take the form of normalised intensity histograms over the pixels within each superpixel and SIFT [14, 16] descriptors of a fixed size calculated at the superpixel centroids.

The location features are based on a local coordinate system for each vertebra. This helps segmentation by providing features that describe the superpixel's relative location. The local coordinates are obtained by matching a contour to the top of the vertebral body. We do this by first (automatically) cropping the ground truth segmentation contours above their centroids, so that the resulting contour set $\mathcal{C}$ corresponds to the upper, roughly semi-circular, boundary of each vertebral body in the ground truth set. Each ground truth image is therefore associated with a single contour $C \in \mathcal{C}$ and our goal is to find the best matching contour of the set for a new image. We use a Laplacian of Gaussian (LoG) filter to detect the outer boundary of the vertebra and search over the image to find the point where the average LoG response along the contour is greatest. The best match is the contour with the maximum response of the set. Features are derived from the matched contour region by centring the pixel coordinates at the region's centroid and computing the gradient of the distance transform [9]. While the matching process depends on the presence of an adequate number of ground truth contours, in practice only an approximate match to the vertebra is required to derive the location features. Using a set of generated synthetic contours is a possibility in cases where the ground truth data is very limited.

Finally, the features in $\mathbf{y}_i^E$ are distinctive of superpixels at the edges and corners of the vertebrae and help to separate the vertebra and background classes around the boundary. We take the LoG response within the superpixel over 4 different scales to form the first feature vector. The second feature vector is formed from the eigenvalues of the structure tensor [12] within the superpixel, taken over 4 scales.

4 Potential Functions

We next describe the potential functions used in (1). Both the data and smoothness terms of the CRF are based on the characteristics learned from superpixel training examples.

We first convert the pixel-level ground truth labels into superpixel-level labels by assigning each superpixel to the class with the majority vote; as Fig. 1 illustrates, there is little ambiguity in this assignment. We then use the superpixel feature/label examples to train a support vector machine (SVM) [5] using an RBF kernel, given by

$$K(\mathbf{y}_i, \mathbf{y}_j) = \exp\left(-\gamma ||\mathbf{y}_i - \mathbf{y}_j||_2^2\right) \tag{3}$$

where γ is a kernel width parameter found using cross-validation on the training data. Probability estimates for the vertebra and background classes are obtained from the SVM using the method of [18] and incorporated into the data term of the CRF. To do this we define the data term as the negative log likelihood of an observation (feature vector) given the class label (i.e. vertebra or background):

$$\psi(\mathbf{y}_i \mid x_i) = -\log\left(P(\mathbf{y}_i \mid x_i)\right) \tag{4}$$

where the likelihood term $P(\mathbf{y}_i \mid x_i)$ for each superpixel is given by the SVM posterior probability. The superpixel likelihoods given by the data term are highly discriminative and localised to the vertebrae regions, as can be seen in the examples shown in Fig. 2b. Note that all pixels within a given superpixel are assigned the same probability, so the figure shows the superpixel-wise probability estimates.

For the second-order potential of our CRF model, we use *distance metric learning* to learn an appropriate distance metric between the superpixel features. While second-order penalties based on standard Euclidean distance measures are often used in graph cut formulations, metric learning tailors the distance measure to the data itself, rather than being chosen *ad hoc*. In particular, we use the Large Margin Nearest Neighbour (LMNN) [17] algorithm to learn a pseudometric of the form

$$D_{\mathbf{M}}(\mathbf{y}_i, \mathbf{y}_j) = (\mathbf{y}_i - \mathbf{y}_j)^\top \mathbf{M}(\mathbf{y}_i - \mathbf{y}_j). \tag{5}$$

The metric is estimated by learning a linear transformation of the data $\mathbf{L}$ such that $\mathbf{L}^\top\mathbf{L} = \mathbf{M}$. The goal is that the k-nearest neighbours of examples in the transformed space (determined by $\mathbf{L}$) should belong to the same class while those belonging to different classes should be separated by a large margin.

We incorporate the learned metric into the second-order potential function as follows

$$\phi(\mathbf{y}_i, \mathbf{y}_j \mid x_i, x_j) = \begin{cases} \exp\left(-D_{\mathbf{M}}(\mathbf{y}_i, \mathbf{y}_j)\right) & \text{if} \quad x_i \neq x_j \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

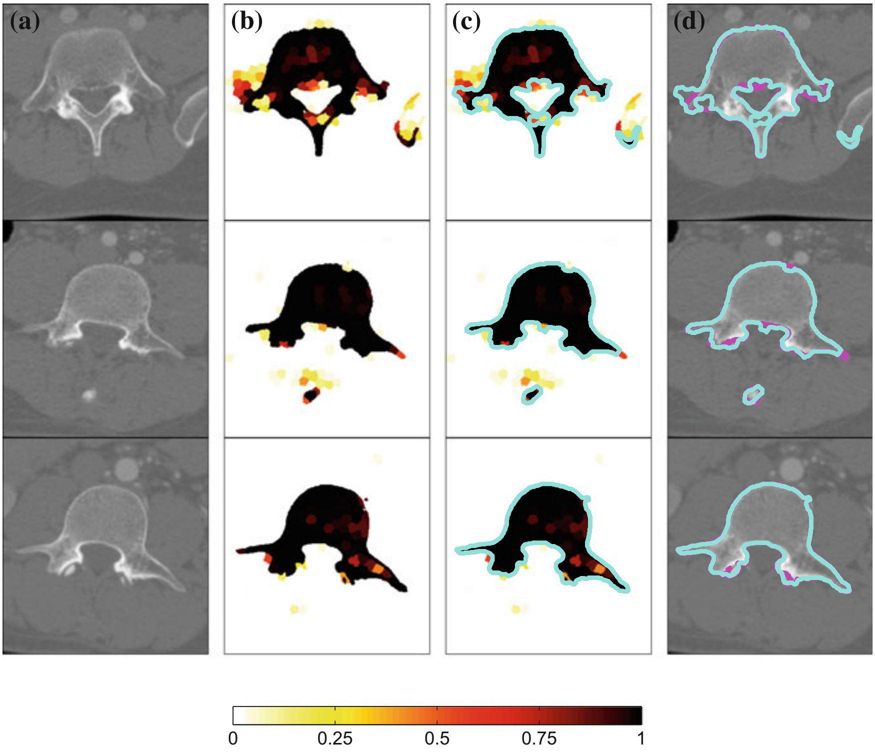

Fig. 2 **a** Shown *top* to *bottom* are CT images corresponding to the minimum, median and maximum Dice similarity score (0.88, 0.97 and 0.98), respectively. **b** SVM probability estimates for the images in the *left* hand column. *Darker* regions indicate higher probability of belonging to the vertebra class. **c** Final segmentation contours from the CRF shown overlaid with the probability estimates (*cyan*). **d** Segmentation contours shown for both the ground truth annotations (*magenta*) and CRF model (*cyan*) (color figure online)

which penalises neighbouring superpixels which have similar feature vectors and are assigned to different classes. The final segmentations using the CRF are compared with the probability estimates from the data term in Fig. 2c.

5 Experiments

We next assess the performance of the method. We first describe the data used for the experiments and the training procedure for the CRF. The segmentation results are then discussed.

5.1 Experimental Setup

The CT data consists of 2D axial slices of lumbar vertebrae from 10 different subjects, each of which has been manually annotated.[1] The images were acquired with Philips or Siemens multi detector CT scanners using an in plane resolution of between 0.31 and 0.45 mm with a slice thickness of 1 mm [19]. We used a total of 50 ground truth images by selecting the middle vertebral slice from each of the 5 lumbar vertebrae of each manually annotated subject. The 512 × 512 pixel images were cropped to 391 × 371 using a bounding box around the vertebrae regions.

The experiments were carried out on a 4-core Intel i5 2.50 GHz machine with 8 GB of RAM. The implementation is written in MATLAB with outside C++ code for certain tasks including superpixel extraction, SVM optimisation and CRF minimisation using graph cuts.

5.2 Model Training

To train the SVMs, leave-one-out (LOO) cross-validation was performed by leaving out one subject (i.e. 5 images) on each iteration and training on the remaining 45 images. The model was then tested on the 5 images from the held out subject and the process was repeated for all 10 subjects. Thus the training and test images were always from separate subjects. The SVM cost parameter $C = 4$ and the kernel width parameter $\gamma = 0.25$ were determined by cross-validation and used for all training runs.

Note that the training data is unbalanced, as there are many more negative (background) examples than positive (foreground) examples. We addressed this by training on a fixed proportion of randomly sampled positive and negative examples. The same LOO approach was used for the LMNN algorithm, with the distance metric learned on the training images for each LOO iteration and applied on the 5 held out images.

5.3 Segmentation Results

To evaluate the degree of overlap between the automatic segmentation and the ground truth, the Dice similarity coefficient (DSC) was used. Given two segmentations $\mathbf{x}$ and $\mathbf{x}'$, the DSC score is defined as

$$\mathrm{DSC}(\mathbf{x}, \mathbf{x}') = \frac{2|\mathbf{x} \cap \mathbf{x}'|}{|\mathbf{x}| + |\mathbf{x}'|}. \tag{7}$$

[1] Data from the CSI2014 segmentation competition is available from the SpineWeb initiative: http://spineweb.digitalimaginggroup.ca.

The score is in the range [0, 1] with 0 indicating no overlap and 1 indicating maximum overlap. LOO testing was used to evaluate the segmentation performance of the method, with the scores taken over all LOO runs.

The segmentations were also evaluated using three distance measures. The mean symmetric absolute surface distance (MSD) score is determined by finding for each set of boundary pixels of both the segmentation and corresponding ground truth, the closest boundary pixels of the other set. The mean of the Euclidean distances to the closest points gives the score for the image, with 0 indicating a perfect segmentation. The RMS symmetric surface distance takes the squared distances between the two sets of boundary pixels, with the final score defined as the root of the average squared distances. Finally, the maximum symmetric absolute surface distance is similar to the MSD score but takes the maximum of the distances instead of the mean. Further discussion of these metrics is provided in [6].

The average processing time for segmentation of a single image was approximately 50 s. The average DSC score was 0.97 with standard deviation 0.01 and the average MSD score was 1.83 with standard deviation 2.54. Table 2 summarises the results obtained on each lumbar vertebra using the evaluation metrics. Figure 2d shows example segmentation contours for both the ground truth and CRF model, corresponding to the minimum, median and maximum DSC score (0.88, 0.97 and 0.98). As the figure suggests, in most cases the automatic segmentation is very close to the manually determined region.

The results obtained by our method compare favourably with those recently presented in [8], who reported an average DSC score of 0.94 ± 0.02. In the same work, the authors showed that their method obtained superior results compared with two other recent approaches to vertebra segmentation [10, 13].

Table 2 Minimum, median and maximum values of the evaluation metrics for each lumbar vertebra

Metric		L1	L2	L3	L4	L5
Dice score	Min	0.92	0.96	0.95	0.94	0.88
	Median	0.97	0.97	0.97	0.97	0.97
	Max	0.98	0.98	0.98	0.98	0.98
Mean surf. dist.	Min	0.91	0.61	0.88	0.85	0.85
	Median	1.20	1.09	1.34	1.37	1.40
	Max	5.29	1.96	1.77	1.99	7.99
RMS surf. dist.	Min	1.38	0.89	1.65	1.30	1.15
	Median	2.00	2.11	2.13	2.52	2.79
	Max	14.87	8.29	4.30	5.47	22.25
Max surf. dist.	Min	5.00	3.17	9.00	6.71	6.00
	Median	14.02	12.39	15.51	14.53	15.62
	Max	91.76	71.87	32.56	42.30	101.55

5.4 3D Segmentation of Vertebrae

The method we have described can also be used to obtain 3D segmentations of vertebrae from individually segmented slices by modifying the way the location features are derived. To do this, the contour matching is first carried out on each slice of the image stack. We then use the M-estimator sample consensus (MSAC) [15] algorithm to remove poor contour matches by detecting and eliminating outliers. Outliers are determined based on the distance to their k-nearest neighbours in the set of matched contours and removed by fitting a polynomial curve through the set of inliers. Location features analogous to the 2D case can then be derived from the correctly matched contours by computing the distance transform in 3D. Figure 3 shows an example 3D vertebra segmentation constructed from segmentations of the constituent slices.

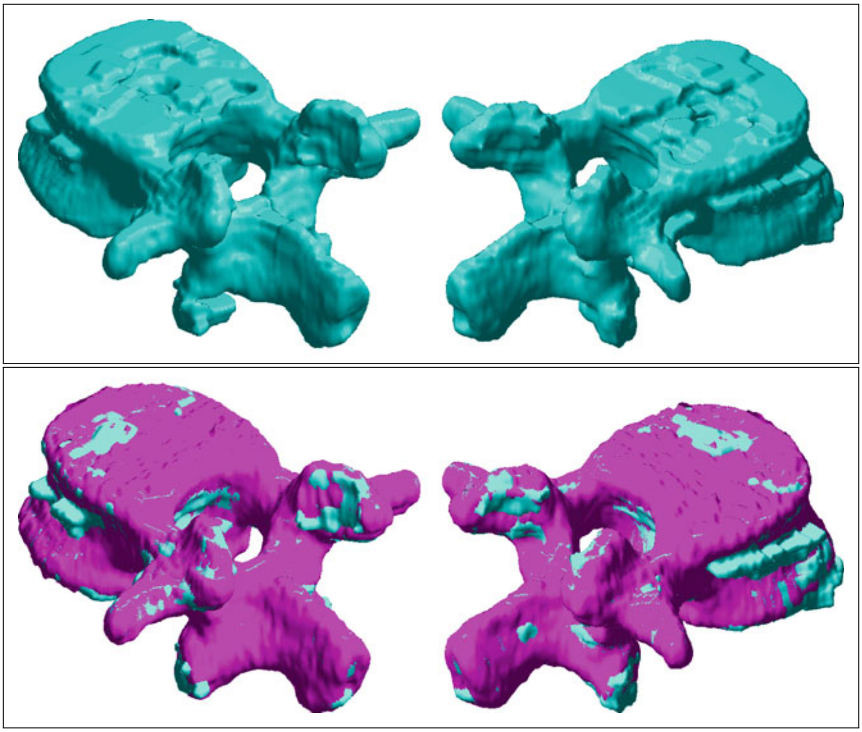

Fig. 3 The *top* figure shows a 3D segmentation of a lumbar vertebra (L2) constructed from segmentations of the constituent slices. The *bottom* figure shows the overlap between the CRF segmentation (*cyan*) and ground truth (*magenta*) (color figure online)

6 Conclusion

We presented an automatic approach for segmentation of vertebra slices from CT images. Our method avoids the requirement of explicit prior shape information and can therefore deal with a wide range of anatomical variation. The results demonstrate that consistently accurate segmentations can be obtained on each of the different lumbar vertebrae from a variety of subjects. Key to the effectiveness of this method is the learning of superpixel features from ground truth data for incorporation into the conditional random field, which in turn ensures spatial coherence. We note that much poorer performance is obtained with traditional features such as just intensity histograms. Finally, we note that this method may be extended to 3D segmentation in a straightforward way. Future work will aim to improve the results in 3D by operating on supervoxels rather than superpixels and by generalising the set of features to characterise the supervoxel regions.

Acknowledgments H. Hutt was funded by the EPSRC. We are grateful to the SpineWeb initiative for making the data available and to the organisers of the CSI2014 competition.

References

1. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Susstrunk, S.: SLIC superpixels compared to State-of-the-Art superpixel methods. IEEE Trans. Pattern Anal. Mach. Intell. **34**(11), 2274–2282 (2012)
2. Blake, A., Kohli, P., Rother, C. (eds.): Markov Random Fields for Vision and Image Processing. The MIT Press, Cambridge (2011)
3. Boykov, Y., Funka-Lea, G.: Graph cuts and efficient N-D image segmentation. Int. J. Comput. Vis. **70**(2), 109–131 (2006)
4. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Trans. Pattern Anal. Mach. Intell. **26**(9), 1124–1137 (2004)
5. Chang, C.-C., Lin, C.-J.: LIBSVM: A library for support vector machines. ACM Trans. Intell. Syst. Tech. 2:27:1–27:27 (2011). Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm
6. Gerig, G., Jomier, M., Chakos, M.: Valmet: a new validation Tool for assessing and improving 3D object segmentation. In: Medical Image Computing and Computer-Assisted Intervention (MICCAI), vol. 2208, pp. 516–523. Springer (2001)
7. Ghosh, S., Alomari, R., Chaudhary, V., Dhillon, G.: Automatic lumbar vertebra segmentation from clinical CT for wedge compression fracture diagnosis. In: SPIE Conference Series, vol. 7963 (2011)
8. Huang, J., Jian, F., Wu, H., Li, H.: An improved level set method for vertebra CT image segmentation. BioMed. Eng. OnLine **12**(48) (2013)
9. Hutt, H. W., Everson, R. M., and Meakin, J. R.: Automatic segmentation of vertebrae from MR images. Technical Report, 2014, School of Physics, University of Exeter (2014)
10. Kim, Y., Kim, D.: A fully automatic vertebra segmentation method using 3D deformable fences. Comput. Med. Imaging Graph. **33**, 343–352 (2009)
11. Klinder, T., Ostermann, J., Ehm, M., Franz, A., Kneser, R., Lorenz, C.: Automated model-based vertebra detection, identification, and segmentation in CT images. Med. Image Anal. **13**(3), 471–482 (2009)

12. Knutsson, H.: Representing Local Structure Using Tensors. In: The 6th Scandinavian Conference on Image Analysis, Oulu, pp. 244–251 (1989)
13. Lim, P.H., Bagci, U., Bai, L.: Introducing willmore flow into level set segmentation of spinal vertebrae. IEEE Trans. Biomed. Eng. **60**(1), 115–122 (2013)
14. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
15. Torr, P.H.S., Zisserman, A.: MLESAC: a new robust estimator with application to estimating image geometry. Comput. Vis. Image Underst. **78**, 138–156 (2000)
16. Vedaldi, A., Fulkerson, B.: VLFeat: an open and portable library of computer vision algorithms (2008). http://www.vlfeat.org/
17. Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. J. Mach. Learn. Res. **10**, 207–244 (2009)
18. Wu, T.-F., Lin, C.-J., Weng, R.C.: Probability estimates for multi-class classification by pair wise coupling. J. Mach. Learn. Res. **5**, 975–1005 (2004)
19. Yao, J., Burns, J. E., Munoz, H., Summers, R.M.: Detection of Vertebral Body Fractures Based on Cortical Shell Unwrapping. In: Medical Image Computing and Computer-Assisted Intervention (MICCAI), vol. 7512, pp. 509–516. Springer (2012)