

Machine Learning Based Bone Segmentation in Ultrasound

Nora Baka^{1(✉)}, Sieger Leenstra², and Theo van Walsum^{1(✉)}

¹ Biomedical Imaging Group Rotterdam,

Department of Radiology & Nuclear Medicine and Medical Informatics,
Erasmus MC, University Medical Center Rotterdam, Rotterdam, The Netherlands

{n.baka,t.vanwalsum}@erasmusmc.nl

² Department of Neurosurgery,

Erasmus MC, University Medical Center Rotterdam, Rotterdam, The Netherlands

Abstract. Ultrasound (US) guidance is of increasing interest for minimally invasive procedures in orthopedics due to its safety and cost benefits. However, bone segmentation from US images remains a challenge due to the low signal to noise ratio and artifacts that hamper US images. We propose to learn the appearance of bone-soft tissue interfaces from annotated training data, and present results with two classifiers, structured forest and a cascaded logistic classifier. We evaluated the proposed methods on 143 spinal images from two datasets acquired at different sites. We achieved a segmentation recall of 0.9 and precision 0.91 for the better dataset, and a recall and precision of 0.87 and 0.81 for the combined dataset, demonstrating the potential of the framework.

1 Introduction

Ultrasound (US) guidance is of increasing interest for minimally invasive procedures in orthopedics due to its safety and cost benefits compared to the more conventional X-ray guidance systems. However, bony structure segmentation from US images remains a challenge due to the low signal to noise ratio and artifacts that hamper US images. In this work we propose to learn the appearance of the bone-soft tissue interface from annotated training data, we propose novel features for robust classification, and show its performance on 143 images from two datasets.

In the literature several heuristic techniques have been published for US bone segmentation. In these works a cost function was manually constructed based on the image appearance. Kowal et al. utilized the fact that bones, after an intensity correction that took the expected depth of the bone into account, have the brightest intensity on the image [8]. This approach works well if the depth of the bone is known and no other tissue interfaces are close by, it is though prone to fail in a less controlled imaging setup. Hacihaliloglu et al. proposed to use phase symmetry of the image in the frequency domain [4]. This approach highlights lines and step edges irrespective of image intensity and is therefore well suited for bones at variable depths. It is though not possible to distinguish bone interfaces

from other bright interfaces. The feature most widely used to detect bone interface is shadowing, which is caused by the reflection of nearly the entire sound wave by the bone. Karamalis et al. [7] proposed a shadow term which was then incorporated in the above framework by Quader et al. [9], showing improved results. Jain et al. proposed a Bayesian framework for bone segmentation, with the following features: intensity, gradient, shadow, intensity profile along scan-line, and multiple reflections [5]. Obtaining the correct conditional probabilities used in the framework is though not straightforward. Foroughi et al. heuristically combined intensity, shadow, and the Laplacian filtered image for creating a bone probability map, which was then segmented with a dynamic programming framework [3]. Jia et al. [6] extended this work by including further feature images such as integrated backscatter, local energy, phase and feature symmetry. All features were normalized and multiplied to derive the bone probability image. As opposed to the heuristic bone probability calculations in literature, we propose to learn the bone probability map from a set of training examples.

Machine learning has been widely used for medical image segmentation, such as in brain MRI, prostate US, etc. Conventionally, such approaches segment a structure with a specific intensity and texture profile, surrounded by a closed contour. Bone interface segmentation in US is different, as here only the outer reflection is visible. Nevertheless, this bone interface has specific features which make it a good candidate for machine learning algorithms. We propose a set of features in this paper and show the accuracy we can achieve with them.

The purpose of this study thus was to answer the following questions. (1) Is it possible to learn a discriminative classifier for bone interfaces in US? (2) Are the features used in the literature robust to different scanning protocols? (3) What accuracy can we achieve with such a system? We address these questions using two 2D spine datasets acquired at different hospitals with different protocols.

2 Methods

The full segmentation framework is shown in Fig. 1. After pre-processing we computed features to discriminate bone from soft tissue interfaces. We used both standard and novel features as described in Sect. 2.2. These features were fed into a classifier to produce a bone probability map. We investigated two types of classifiers, namely pixel-wise and region based classifiers. As a post-processing step dynamic programming was used to produce the final segmentation.

2.1 Pre-processing

We applied two pre-processing steps. First, a Gaussian blurring produces the blurred image I_σ . We empirically found that a standard deviation of $\sigma = 0.3$ mm worked well, smoothing the speckle pattern but still allowing to distinguish adjacent tissue interfaces. A typical speckle size in the depth direction of our images was about 0.1 mm. Second, oblique edges appear with less contrast in the images as only part of the reflected sound reaches the transducer. We propose to enhance

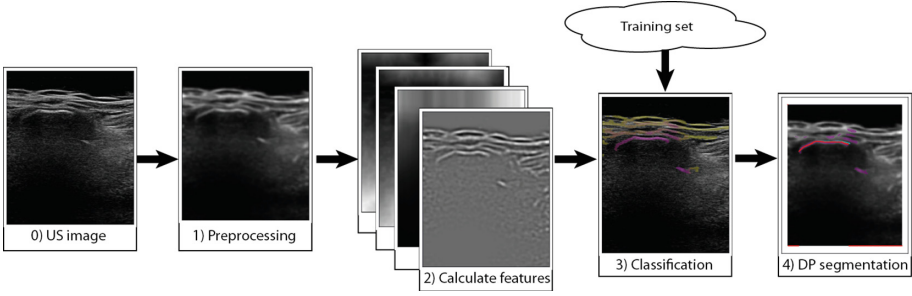


Fig. 1. The schematic method for classification based bone segmentation.

oblique regions with template matching. We took as template the Gabor filter with orientations of 45° and -45° , wavelength $\lambda = 2$ mm, and Gaussian width $\sigma_g = 1$ mm. These parameters were set such that the filter resembles a single oblique edge with some room for deviation from the above angles. The filtered images were then thresholded for retaining only the high score regions, and added to the blurred image such that

$$I_{pre} = I_\sigma + \delta F(Gabor(\lambda, \sigma_g, 45^\circ) * I_\sigma) + \delta F(Gabor(\lambda, \sigma_g, -45^\circ) * I_\sigma), \quad (1)$$

where I_{pre} is the final pre-processed image, and F is representing a thresholding with 2, and a subsequent Gaussian blurring with σ . The threshold value was selected empirically, such that only the strongest responses remained, and the subsequent blurring was used to smooth the edges. We used an $\delta = 10$ weighting, such that the maximum value of the enhancement was about 50 for an image with maximum intensity of 255. Examples of original and pre-processed images are shown in Figs. 1 and 2.

2.2 Features

We used standard bone features from the literature, namely

1. Intensity of the pre-processed image I_{pre} .
2. Laplacian of Gaussian (LoG) image $I_{LoG} = -(LoG * I_{pre})$, with scale σ .
3. Intensity shadow, which was calculated as the cumulative sum of image intensities from bottom to top, scaled with the pixel size of the image.
4. Depth. This feature is the distance in mm-s from the transducer, and is mainly used to discard the structures too close to the transducer.

In addition to these standard features, we defined the following novel features.

5. Border-to-border distance. This feature discriminates bright lines based on their length. Lines that run from the left edge of the field of view (FOV) all the way to the right edge will have a lower value than shorter lines and areas without line-like structures. We use the Laplacian of Gaussian image

for enhancing bright line structures. Subsequently, we calculate the weighted distance function from the left border and the right border of the image respectively, where the weight image is $I_{weight} = 1/(\max(0, I_{LoG}) + 0.01)$. The sum of the left and right distances results in low values for pixels that participate in a long structure, and higher values for short lines. It is useful if we know that the imaged bone width is less than the FOV, which is the case with most bones.

6. Centrality. This feature urges the classifier to find structures that are closer to the center column of the field of view, as those structures tend to be of greater importance than the ones on the border. We calculate this feature as the weighted distance function from the center column of the image in both directions. The same weight image I_{weight} was used as in the border-to-border distance.

All feature images except depth were normalized by division by their maximum value. Depth was normalized by division by 30, so that depth values were between 0–2. In addition, the shadow feature images were translated 1 mm to the top, in order to sample the cumulative shadow without the possibly bright bone interface. An example showing all feature images can be seen in Fig. 2.

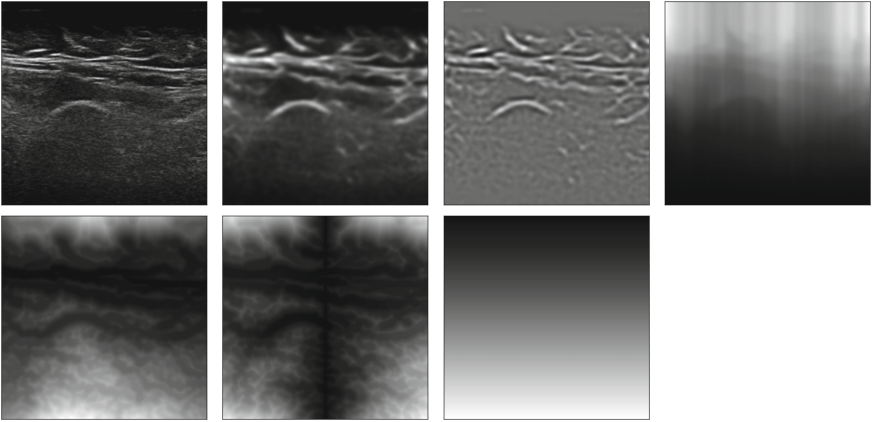


Fig. 2. Feature images from left to right: Original image, pre-processed image, LoG image, intensity shadow, border-to-border distance, centrality and depth.

2.3 Classifiers

We compared two classifiers in this work. For the pixel-wise classification we chose the logistic classifier, as more flexible classifiers quickly overfit. For region based classification we chose structured forests, as it showed promising results on edge detection in natural images [2]. The following paragraphs describe both classifiers in detail.

Logistic Classifier. To make better use of the training data, we implemented a cascade classifier scheme. In the first step, we discarded all regions that did not contain structure, by a hysteresis threshold $HysThr(I_{LoG}, 0.2, 0.03)$ of the normalized I_{LoG} feature. In the second step, a logistic classifier was trained on the pixels remaining after the first step. Negative samples were collected from locations passing the first cascade step, and being further than 2 mm from the ground truth. We sampled all positive samples (typically 200–400 samples), and 5000 negative samples from each training image. We then used weighting to balance the influence of both classes on the decision boundary.

Structured Forest. The structured forest (SF) classifier introduced for edge detection in natural images by [2] was used in this work. It is a random forest classifier where the input and output are patches rather than pixels. We replaced the original RGBD input channels with the feature images described in Sect. 2.2. The feature pool is calculated from these channels by sampling pixels from the patch, and by calculating the difference of any two pixels in the patch after down-sampling it to a size of 5×5 . The 6 channels with a patch size of 32 pixels thereby produce 3336 feature candidates for the forest. The intermediate mapping used for the splitting criterion of the tree nodes works best if areas under, and above the bone interface have different labels. We therefore dilated the ground truth downwards with 5 pixels to form the foreground segmentation. Training patches were only sampled from locations where $I_{LoG} > 0.3$ to avoid sampling locations with no structure. We trained a forest of 4 trees and three scale levels, using $1e5$ positive and $2e5$ negative samples. All further parameters were set as proposed in [2]. The output image that results after applying the tree to an unseen US image was blurred according to the scale, and added up to result the forest output. To counteract the intensity loss due to spacial blurring, we then normalized the image to have the highest probability equal to 100.

2.4 Dynamic Programming Segmentation

The classification step produces a probability map of the expected bone interface in the image. To get a final smooth segmentation, we use this probability map and dynamic programming similar to [3], with the cost function $C = C_{intern} + C_{extern}$. The internal energy is created as

$$C_{intern} = 1 - P_{bone}, \quad (2)$$

with an extra line added to the cost image with value 0.6. The segmentation is at this line if there is no bony structure in the image column. The external energy consists of the penalty for moving between columns, as follows:

$$C_{extern} = \begin{cases} jumpCost, & \text{if jump from or to extra line} \\ \alpha(i_{rowCurr} - i_{rowNext})^2, & \text{with maximum 3 rows difference} \end{cases} \quad (3)$$

We used $jumpCost = 1$ and $\alpha = 0.1$ in our experiments.

3 Data

We used two sets of 2D ultrasound images in this study.

Dataset 1 consisted of a training set of 106 vertebra images of 15 subjects (Dataset 1A) and a test set of 56 images of 10 subjects (Dataset 1B), acquired at the Tilburg Hospital with a 2D Philips L12-3 Broadband linear array transducer. Every vertebra was imaged for about 3 sec at a gain of 45, contrast 55, and base imaging depth of 5 cm. This depth was adjusted if needed. The patient was imaged in a supine position. For every vertebra one image of the sequence was selected for manual annotation of the ground truth bone surfaces.

Dataset 2 consisted of a training set of 91 vertebra images of 11 subjects (Dataset 2A) and a test set of 87 images of 10 subjects (Dataset 2B), acquired at Erasmus MC, with a Philips iU22 machine and the 2D L12-5 linear transducer. The images were acquired with the general musculo-skeletal protocol, with imaging depth adjusted between 3–5 cm depending on the patient, and gain 65. Focus was adjusted so that the imaged bone should be in the focus region. Pixel size in most cases was around 0.1 mm. SonoCT and XRES adaptive image enhancement were switched on. The patient was imaged in a sitting position. The patient population consisted of back-pain patients of ages between 19 and 77, with BMI between 17.9 and 39.1. For every vertebra one image of the sequence was selected for manual annotation of the ground truth bone surfaces.

Manual annotation was done by spline interpolation between manually placed control-points in both datasets. In every image only one vertebra was annotated. In dataset 1 due to the smaller field of view of the transducer this meant all visible bony structures were segmented. In dataset 2 with a larger field of view in about half of the cases the edge of neighboring vertebral processes were also visible, these were not included in the ground truth segmentation.

4 Experiments and Results

We performed experiments to (a) evaluate the accuracy of the two classifiers and their segmentation performance; (b) assess the robustness of the methods to different acquisition setups; (c) compare the method with the method of Foroughi [3]. Comparison with Foroughi was only performed on dataset 1, as there all bony structures were segmented in the ground truth. To evaluate the performance of the classifiers and the subsequent dynamic programming segmentation, we used recall, precision, and the F-measure as follows:

- Classifier recall (Rec_C): The ratio of ground truth contour that was classified correctly and the length of the entire contour.
- Classifier precision ($Prec_C$): The ratio of detection inside the dilated ground truth compared to all detections.
- Segmentation recall (Rec_S): the number segmentation points inside the dilated ground truth region, divided by the number of ground truth pixels. As every column has maximum one pixel marked as bone interface, this results in a normalized number, such that the perfect contour has sensitivity 1, and if

bone was not found, sensitivity is 0. However, sensitivity might also be larger than 1 due to the region enlargement. Values larger than 1 are thresholded to 1.

- Segmentation precision ($Prec_S$). The length of the segmentation inside the dilated ground truth divided by the total length of the segmentation.
- F-measure: The F-measure is the harmonic mean of precision and recall: $F = 2 \frac{prec \cdot rec}{prec + rec}$. The combination of these two measures facilitates the comparison of classifier accuracies.

Dilation for the ground truth was set to 2 mm where dilation was used.

The threshold value used to calculate the classification evaluation measures was set to 50% for the logistic classifier, and to 30% for the structured forest classifier. These values were optimized based on the training set. For the dynamic programming the parameter values are mentioned in Sect. 2.4. These parameters were optimized on the Foroughi method with training set A, and were used for all experiments in this paper.

The results of the experiments are shown in Table 1. Besides the average precision and recall we also report the number of failures, defined by a segmentation precision or recall < 0.01 . Figure 3 shows examples of bone probability images and their segmentation using the relative shadow feature.

Table 1. Classification and segmentation results of the proposed framework on the test set, together with results of the method of Foroughi et al. [3]. 1 and 2 denote the two different ultrasound datasets used in this paper, 1A2A and 1B2B denotes the combined training and test dataset respectively.

Classifier	Train	Test	Segmentation			Classification			
			Recall	Precision	F-meas	F-meas	Recall	Precision	#fail
Logistic	1A	1B	0.93	0.76	0.84	0.77	0.87	0.69	1
Logistic	2A	2B	0.93	0.59	0.72	0.70	0.87	0.59	1
Logistic	1A	2B	0.85	0.39	0.53	0.45	0.89	0.30	6
Logistic	2A	1B	0.62	0.61	0.62	0.63	0.56	0.71	12
Logistic	1A2A	1B2B	0.89	0.57	0.69	0.64	0.87	0.50	7
SF	1A	1B	0.90	0.91	0.90	0.85	0.84	0.87	0
SF	2A	2B	0.83	0.80	0.80	0.78	0.77	0.79	6
SF	1A	2B	0.78	0.61	0.66	0.65	0.76	0.57	10
SF	2A	1B	0.58	0.69	0.61	0.62	0.58	0.67	8
SF	1A2A	1B2B	0.87	0.81	0.82	0.80	0.82	0.79	4
Foroughi		1B	0.71	0.67	0.66				6

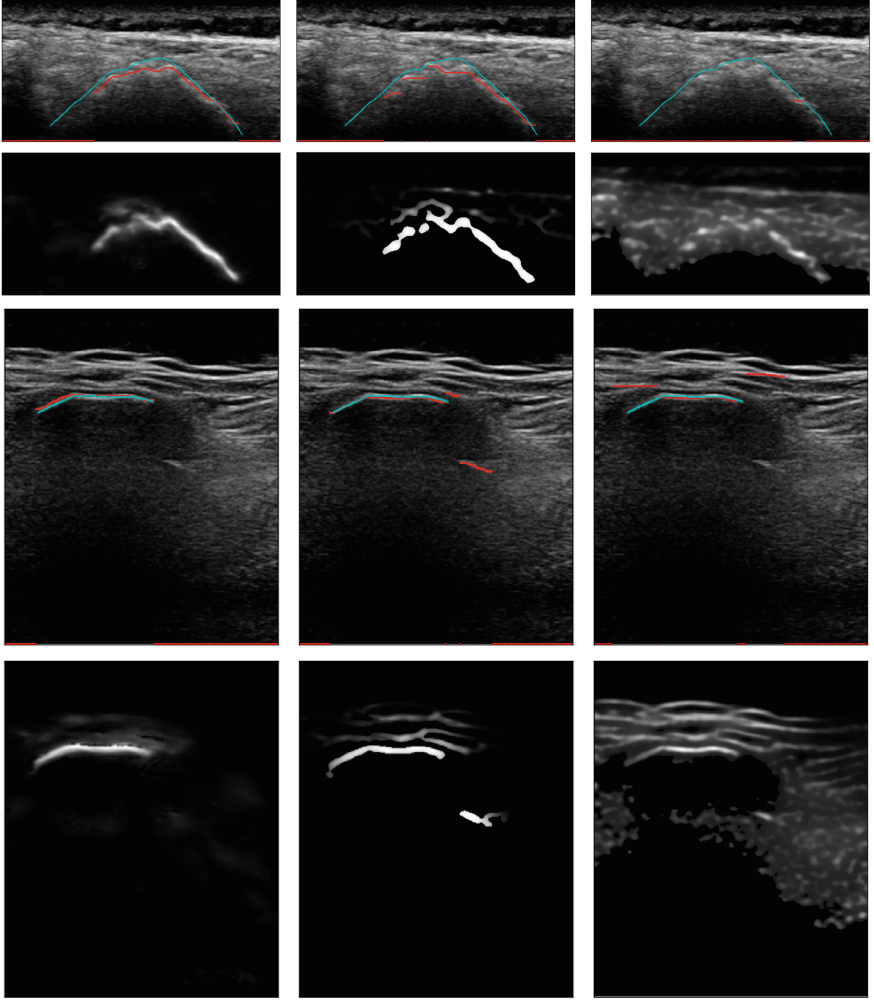


Fig. 3. Example images and segmentations from dataset 1. From left to right: Structured forest, Logistic classifier, and Foroughi et al. [3]. The original images with ground truth (blue) and dynamic programming segmentations (red) are in the odd rows. The corresponding probability images are in the even rows. (Color figure online)

5 Discussion and Conclusions

We performed experiments evaluating bone interface classification and segmentation from US images in two datasets (Table 1). The two datasets were acquired on different machines and with different protocols. Though visually the images looked similar in the two datasets, the low cross-evaluation accuracy (F measure between 0.45 and 0.65) shows that the statistical properties of the two datasets were different. The largest difference we found in the histogram of the shadow

feature, which showed a clear difference in the optimal threshold value for separation of bone and non-bone interfaces in the two datasets.

We also found that structured forest with the relative probability threshold consistently achieved higher accuracy than the logistic classifier (with a difference between 8–20 % F-measure). Notably, the SF classifier on the combined dataset achieved an accuracy close to the average accuracy of the purely trained classifiers. This might be due to the selection of the best features during training, the non-linear decision boundary, and the use of regional information. This result is important, as it suggests, that a smart classifier may compensate for statistical differences in datasets, thereby facilitating robust generally applicable methods.

We also looked at the failure cases in the combined dataset experiment. The structured forest classifier failed on four cases. All four cases were from dataset 2. Two failures were images of short bones far from the FOV center. These were not detected. One image contained two vertebrae. As only one of them was annotated in the ground truth, and the other one was segmented by the method, it counted as a complete failure. The last failure was due to a wrong FOV cropping. There were 80 zero-padded columns on the right of the image, which interfered with the way the border-to-border feature is calculated. In lack of this feature the fat-muscle interfaces were segmented instead of the bone-soft-tissue interface. Generally, bright well visible bones with distinctive shadow are segmented with the highest confidence.

Comparing segmentation and classification performances, segmentation may outperform classification. This is possibly due to the added smoothness and spatial connectedness constraints that help to ignore small false positive and negative classifications. All dynamic programming segmentation experiments were performed with the same parameters, optimized for the method of Foroughi [3]. Further improvements might be possible with specially tuned parameter values for the different datasets.

We also compared our results with the method of Foroughi et al. [3]. Both the logistic and structured forest classifiers outperformed this heuristic approach. We also conclude that our dataset must be more challenging than the dataset used in the original paper, as the results are substantially worse than the reported ones.

The computation time of the structured forest, with an implementation based on the optimized Matlab code of [1,2], took 0.5 s, and the subsequent dynamic programming another 0.2 s. The logistic classifier was implemented in Python 2.7 using scikit-learn package without speed related optimization. The classifier including the dynamic programming thereby took 2.6 s to compute on a laptop with intel i7-2760QM 2.4 GHz cpu and 16 GB ram.

We conclude that machine learning is a feasible and accurate way for bone segmentation in ultrasound. We achieved best results with the structured forest segmentation scheme with recall and precision as high as 0.90/0.91 on dataset A, and 0.87/0.81 on the combined dataset.

Acknowledgement. The authors were financially supported by the Dutch Science Foundation STW, project number 14542. The authors would furthermore like to thank Marcel Toorop and Eelke Bos for their help in data annotation.

References

1. Dollár, P.: Piotr's Computer Vision Matlab Toolbox (PMT) (2016)
2. Dollár, P., Zitnick, C.L.: Structured forests for fast edge detection. In: 2013 IEEE International Conference on Computer Vision, pp. 1841–1848. IEEE, December 2013
3. Foroughi, P., Boctor, E., Swartz, M.J., Taylor, R.H., Fichtinger, G.: Ultrasound bone segmentation using dynamic programming. In: 2007 IEEE Ultrasonics Symposium Proceedings, pp. 2523–2526. IEEE, October 2007
4. Hacıhaliloğlu, I., Abugharbieh, R., Hodgson, A., Rohling, R.: Bone segmentation and fracture detection in ultrasound using 3D local phase features. In: Metaxas, D., Axel, L., Fichtinger, G., Székely, G. (eds.) MICCAI 2008. LNCS, vol. 5241, pp. 287–295. Springer, Heidelberg (2008). doi:[10.1007/978-3-540-85988-8_35](https://doi.org/10.1007/978-3-540-85988-8_35)
5. Jain, A.K., Taylor, R.H.: Understanding bone responses in B-mode ultrasound images and automatic bone surface extraction using a Bayesian probabilistic framework. In: Walker, W.F., Emelianov, S.Y. (eds.) SPIE Medical Imaging, pp. 131–142. International Society for Optics and Photonics, April 2004
6. Jia, R., Mellon, S.J., Hansjee, S., Monk, A.P., Murray, D.W., Noble, J.A.: Automatic bone segmentation in ultrasound images using local phase features and dynamic programming. In: 2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI), pp. 1005–1008. IEEE, April 2016
7. Karamalis, A., Wein, W., Klein, T., Navab, N.: Ultrasound confidence maps using random walks. *Med. Image Anal.* **16**(6), 1101–1112 (2012)
8. Kowal, J., Amstutz, C., Langlotz, F., Talib, H., Ballester, M.G.: Automated bone contour detection in ultrasound B-mode images for minimally invasive registration in computer-assisted surgery—an in vitro evaluation. *Int. J. Med. Robot. Comput. Assist. Surg.* **3**(4), 341–348 (2007)
9. Quader, N., Hodgson, A., Abugharbieh, R.: Confidence weighted local phase features for robust bone surface segmentation in ultrasound. In: Linguraru, M.G., et al. (eds.) CLIP 2014. LNCS, vol. 8680, pp. 76–83. Springer, Cham (2014). doi:[10.1007/978-3-319-13909-8_10](https://doi.org/10.1007/978-3-319-13909-8_10)

Computational Methods and Clinical Applications for
Spine Imaging

4th International Workshop and Challenge, CSI 2016,
Held in Conjunction with MICCAI 2016, Athens, Greece,
October 17, 2016, Revised Selected Papers

Yao, J.; Vrtovec, T.; Guoyan, Z.; Frangi, A.; Glocker, B.;
Shuo, L. (Eds.)

2016, X, 147 p. 60 illus., Softcover

ISBN: 978-3-319-55049-7