

Fully automatic detection of the vertebrae in 2D CT images

Franz Graf^a, Hans-Peter Kriegel^a, Matthias Schubert^a, Michael Strukelj^a,
Alexander Cavallaro^b

^aInstitute for Informatics, Ludwig-Maximilians-Universität München,
Oettingenstr. 67, 80538 Munich, Germany

^bImaging Science Institute Erlangen, Maximiliansplatz 1, 91054 Erlangen, Germany

ABSTRACT

Knowledge about the vertebrae is a valuable source of information for several annotation tasks. In recent years, the research community spent a considerable effort for detecting, segmenting and analyzing the vertebrae and the spine in various image modalities like CT or MR. Most of these methods rely on prior knowledge like the location of the vertebrae or other initial information like the manual detection of the spine. Furthermore, the majority of these methods require a complete volume scan. With the existence of use cases where only a single slice is available, there arises a demand for methods allowing the detection of the vertebrae in 2D images. In this paper, we propose a fully automatic and parameterless algorithm for detecting the vertebrae in 2D CT images. Our algorithm starts with detecting candidate locations by taking the density of bone-like structures into account. Afterwards, the candidate locations are extended into candidate regions for which certain image features are extracted. The resulting feature vectors are compared to a sample set of previously annotated and processed images in order to determine the best candidate region. In a final step, the result region is readjusted until convergence to a locally optimal position. Our new method is validated on a real world data set of more than 9 329 images of 34 patients being annotated by a clinician in order to provide a realistic ground truth.

Keywords: Annotation, Registration, Similarity Search, Detection, Vertebrae, CT

1. INTRODUCTION

Knowledge about the vertebrae is a valuable source of information for several applications. An example is slice registration where the shape of the spine is a strong indicator of the displayed body region. Another example is spine reconstruction using the extracted information to reconstruct 3D models of the spine. To conclude, information extraction in the field of detecting, segmenting, analyzing and reconstructing meshes of the vertebrae and the complete spine has attracted a lot of attention in the field of medical image processing. However, most of the research in this area is based on the availability of a full 3D volume of CT or MR scans.

Vrtovec et al. construct 3D shape models of the spine and analyze the spinal curvature^{1,2} in CT images while Stern et al. determine the spinal centerline in both CT and MR volume scans.^{3,4} Furthermore, there are several methods aiming at the detection and segmentation of the spine⁵⁻⁷ or the spinal cord and the spinal canal in 3D CT scans by using deformable fences or models.⁸ Approaches that are dealing with 3D MR scans are proposed by Schmidt et al.,⁹ Corso et al.¹⁰ and Huang et al.¹¹ A major disadvantage of the above methods is that they are all dependent on the availability of extended volume scans. While processing the complete scan in order to extract new information often yields good results, there exist use cases where only very small volumes or even single slices of a scan are available. For example, the available slices are taken from a medical record being send to a specialist without access to the complete data in the PACS.

In this paper, we propose a new method for detecting the position of the vertebrae in 2D CT slices on the transverse plane. Thus, especially in applications where the volume of the available scan is rather small or even a single slice, our method generates valuable information which can be employed for navigation and annotation. For example, within the 2D slice the vertebrae is important as an orientation point for determining the center of the body w.r.t. the particular position of the patient on the table. Furthermore, it is possible to classify the detected vertebrae on the image into cervical-, thoracic- or lumbar spine area in order to determine the body

Further author information: graf@dbs.ifi.lmu.de, Telephone: +49 89 2180 9329

region in which the available volume is placed in. Another use of our detector is as a fully automatic preprocessing step for constructing a 3D model of the spine within the available scan volume.

Our new method has several characteristics allowing a broad use in various applications. First of all, our method allows reliable vertebrae detection without time-consuming parameter tuning. Since our method is fully automatic and parameterless, it is not necessary to adjust the parameters to a particular volume before receiving usable results. Furthermore, the detector is based on a directed readjustment step which avoids checking for the target region at any possible position on the 2D slice. As a result, the vertebrae is detected very efficiently allowing to apply the detector on large data sets or in interactive applications. A final useful characteristic of our approach is its low memory consumption making the detector a suitable component of more sophisticated imaging systems.

Technically, our method employs four steps to determine the position of the spine. The first step, comprises several typical preprocessing steps. Then, we detect possible candidate locations of the vertebrae by extracting interesting pixels from a bone density map. Afterwards, image features are extracted for the candidate regions surrounding the interesting pixels and compared to an annotated sample set in order to select the most promising candidate region for further processing. After the best candidate region is identified, the result region is iteratively readjusted in a refinement step until the method converges to a local optimum.

The rest of the paper is organized as follows: The algorithm will be described in Section 2, followed by the description of our experimental setting and the achieved results in Section 3. Section 4 concludes the paper and gives an outlook to future work.

2. LOCALIZATION OF THE VERTEBRAE

In this section, we describe our new algorithm for detecting the vertebrae within a single 2D slice of a CT scan. To determine the most relevant region within the a given image, four consecutive steps are performed which will be described in the following subsections. After describing the basic steps of the algorithm, we additionally discuss some techniques for improving the processing time.

2.1 PREPROCESSING

The first step performs several preprocessing steps to make the relevant information easier to detect. Thus, we reduce the image to the region of interest by cutting of empty borders. A border is defined as block of at least 20 lines (columns) which contains at least 100 continuous pixels above -600 Hounsfield Units (HU). After detecting the borders on all sides of the body, we define the region of interest as the outline of the joined borders. Afterwards, the image is scaled to a size of 512 to 512 pixels and all previously applied windowing is removed. In order to attenuate the effect of noise pixels which are very likely in scans with a high kVp value, a Gaussian kernel of size 5×5 pixels and $\sigma_{x,y} = 1$ is applied to the image. Fig. 1 shows the result of the preprocessing step.

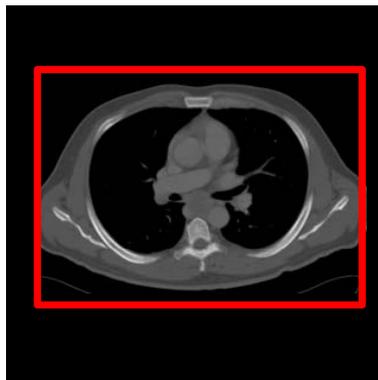


Figure 1. Image after preprocessing and body detection.

2.2 CANDIDATE SELECTION

In the next step, a so-called bone density map is created which is used to identify interesting image regions that might contain the vertebrae. Therefore, we start by determining the set of bone pixels $B \in [1..512] \times [1..512]$ by selecting pixels having a value in a range of 500 and 1000 HU. For each bone pixel, we determine the so-called bone density of a pixel by summing up the Euclidean distances to all other bone pixels of the image w.r.t. the pixel coordinates. Formally, the bone density of a pixel p at position (x, y) is defined as follows:

$$D(x, y) = \sum_{(i,j) \in B} \sqrt{(x-i)^2 + (y-j)^2}$$

The bone density map is constructed by adding an entry for each bone pixel in combination with its bone density. Afterwards, the bone density map is ordered in ascending order w.r.t. the bone density. In other words, the bone pixels denoting the highest bone density are sorted on top of the bone density map.

In order to obtain candidate regions from the bone density map, the first pixel of the list is expanded to a region of 84×68 mm which is twice the average size of an annotated spine region in the sample database of annotated images. We use twice the size of the box because the annotation comprises only the vertebrae whereas we are interested in the vertebrae including the spinous process as well.

Regarding the shape of a vertebra, the highest bone density is expected in the part of the vertebral body. As we want to extract image features describing the vertebral body as well as the spinous process, we need to expand the box not equally in all directions. Instead we expand the box's width in both directions but the height only downwards in order to raise the chance that the box also covers the spinous process. An illustration about the expansion of the box is shown in Figure 5 in Section 3 where the initial point of the bone density map is located at the point in the center of the inner rectangle which represents the average annotation box.

Afterwards, all other entries in the surrounding of 42×34 mm (which is the size of the average annotation) of the top entry are removed from the list in order to avoid the generation of regions being too similar to each other. This step is repeated five times in order to obtain the five most promising candidate regions from the bone density map. Figure 2(a) illustrates the considered bone pixels and Fig. 2(b) displays the selected top 4 candidate boxes.

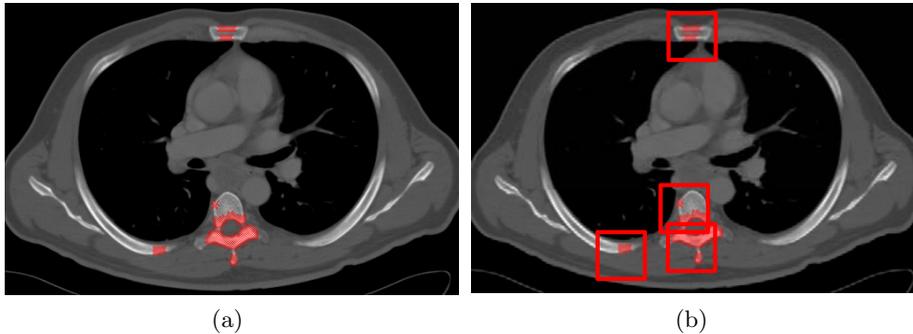


Figure 2. Bone density map (a) and candidates selected from the bone density map (b).

2.3 FEATURE-BASED REGION PREDICTION

In the next step, we derive a feature descriptor for each of the remaining five candidate regions. In order to consider the spatial distribution, a candidate region is divided into nine disjoint, equally sized sub regions. For each of these sub regions, a feature descriptor is derived and the descriptor for the complete region consists of the concatenation of the descriptors of its nine sub regions. In this work, we experimented with three types of image descriptors, i.e. greyscale (HU value) histograms, Haralick texture features and pyramidal histograms of oriented gradients (PHoGs). The performance of each of these descriptors is compared in the next section.

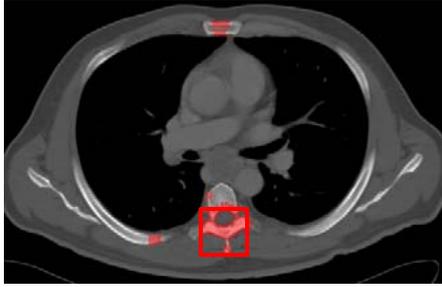


Figure 3. Candidate region selected by the feature-based prediction step.

HU Histograms In the case of HU histograms, we distinguished 16 uniform intervals in the scale of Hounsfield Units (HU). For each bin in the histogram, we counted the number of pixels having a HU value in the corresponding interval. Thus, we have 16 values for each sub region and $16 \cdot 9 = 144$ dimensions for the complete descriptor of a sub region.

Haralick Texture Features As we want to match against image patterns of the vertebrae, we also tested the algorithm with the well known Haralick texture features¹² which also proved valuable information in.¹³ For our proposed method, we compute all 13 Haralick features for five different distance values (1, 3, 5, 7, 11). This computation is done for each of the 9 subregions separately. The resulting feature vector contains $9 \cdot 13 \cdot 5 = 585$ features. As stated in Haralick et al.,¹² some of the features are highly correlated. This means that the resulting feature vector contains a lot of redundant information in its full representation. In order to minimize these redundancies, we apply principal component analysis (PCA) and thus reduce the dimensionality of the texture descriptor to 30 dimensions.

Histograms of Oriented Gradients To extract the gradient features for a candidate region, we extracted important edges P_{edge} by applying the Canny operator¹⁴ C . Important edges are defined by all locations, where the Canny operator computes values greater than zero (1). In the next step, we compute the gradient’s angle $G(x, y)$ at the locations of important edges (2).

$$P_{\text{edge}} = \{(x, y) | C(x, y) > 0\} \quad (1)$$

$$G(x, y) = \arctan \frac{\partial y}{\partial x} ; \text{ where } (x, y) \in P_{\text{edge}} \quad (2)$$

Afterwards, a 12 bin histogram is built over all values of $G(x, y)$ within the complete candidate region and for each of its nine sub regions. The histograms are then serialized generating a $(9 + 1) \cdot 12 = 120$ dimensional vector. This representation is referred to as PHoG (pyramid histograms of oriented gradients).

Candidate Selection To determine the most promising of the five selected candidate regions, we employ a sample set of manually annotated regions containing the vertebrae. Each sample in the database is described by the same feature descriptor as the candidate region. To compare the d -dimensional feature descriptors we employ Manhattan distance: $D_{\text{Manhattan}}(\vec{u}, \vec{v}) = \sum_{i=1}^d |u_i - v_i|$. The most promising candidate region is determined by calculating the nearest neighbor of each candidate in the sample set. The candidate region having the smallest Manhattan distance to its nearest neighbor in the sample set, is selected as result region (cf. Fig. 3).

2.4 REFINEMENT

After selecting the best of the initial candidate regions, we proceed with an iterative optimization step moving the result to an local optimal position. In each step, we generate new candidate regions by moving the position of the currently best candidate region by 5 mm to the top, bottom, left and right. For each of these new candidates, we also determine the distance to the nearest neighbor in the sample set and select the one with the smallest

Manhattan distance. The algorithm terminates, if there is no new candidate region having a smaller distance to its nearest neighbor than the result region in the previous step. The Figure 4 shows the result region before (cf. Fig. 4(a)) and after (cf. Fig. 4(b)) the refinement step.

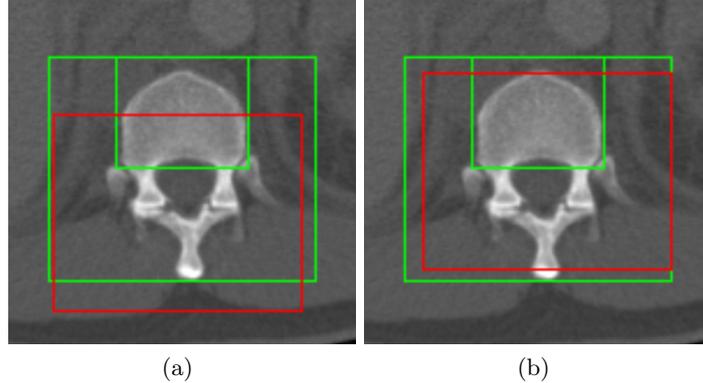


Figure 4. Image before (a) and after (b) the refinement step. The green boxes denote the (doubled) annotation and the red boxes indicate the current result box of the algorithm.

2.5 PERFORMANCE TUNING

In order to improve the processing speed of the algorithm, it is important to accelerate the computation of the bone density map, as it has a runtime of $O(n + m^2)$ with n being the amount of pixels of the image ($512 \times 512 = 262144$ in our case) and m being the amount of pixels indicating bone structure. One way to improve the calculation speed is to rescale the image by a factor of 0.5 first. This way the amount of ordinary pixels and bone pixels can be significantly reduced. A further optimization employs the observation that the analyzed images are recorded while patients are lying on their back. Thus, we can assume that the vertebral column is located in the mid third of the image. Thus, the processing time is reduced to $O(\frac{1}{12}n + \frac{1}{16}\delta^2m^2)$ with $\delta \in [0; 1]$ denoting the amount of bone pixels within the mid third of the image compared to the total amount of bone pixels.

3. EXPERIMENTAL VALIDATION

3.1 Data Set

In our experiments, we use real world thorax CT scans of 34 different male and female patients with a total of 9239 images consuming more than 10 GB disk space. The scans were recorded with different CT scanners with different resolutions along the z-axis and thus, our scans are very heterogeneous w.r.t. resolution (along the x- and y-axis), contrast media, patient’s body size, gender and age. Some patients also show signs of implants which may cause disturbances on the image.

In order to obtain a reliable ground truth, the data set was annotated by a clinician using the MEDICO tool.¹⁵ For the ground truth, each vertebral body of the spine was annotated by a single 3D paraxial bounding box enclosing the vertebral body and a tag that identified the vertebral body (C1-C7, T1-T12, L1-L5). The spinous processes were not included into the annotation boxes because this would have created comparatively large bounding boxes and also quite large overlaps between the annotations of consecutive vertebrae especially in the area of the lumbar vertebrae. Even though that only the vertebral bodies were annotated by enclosing boxes, it was not possible to completely avoid an overlap between annotation boxes. This can be explained by the fact that the annotation boxes are paraxial but the spine describes a curve along the z-axis of the body. Thus, the vertebrae might be rotated to a certain degree.

3.2 Evaluation

In total, we had 393 bounding boxes with sizes varying from 22.9×17.2 mm (36×29 px) to 89.6×53.7 mm (204×120 px) and an average size of 43×36 mm. In the experiments, we conducted a cross validation on the patients’

scans, so that we could ensure that the query image was never compared with other images of the same scan of this patient in order to avoid a bias towards the patient from which the query image was obtained.

In order to evaluate the performance of the proposed algorithm and the tested features, we define an overlap O between the manually annotated box B_m with the box which is determined by the presented algorithm B_a . The intuitive definition of an overlap ($O_{simple} = (area(B_m) \cap area(B_a)) / area(B_m)$) does not cope with the problem that the annotation boxes are varying in size. The vertebral bodies are smaller in the upper part of the spine than in the area of the lumbar spine. In contrast to the varying size of the annotated boxes, the size of the result regions displays a fixed width. The idea to use the largest annotated box was rejected due to the large maximum values encountered in the annotation (89.6×53.7 mm) which would lead to impractical results.

Thus, using the average box size as target size of the algorithm led to the problem that O_{simple} could never reach a value of 100% in regions being larger than the annotation even if the found region was overlapping the annotation perfectly. To solve this problem, we defined the overlap by the ratio of the intersecting area relatively to the smaller of both areas (3):

$$O = \frac{area(B_m) \cap area(B_a)}{\text{argmin}(area(B_m), area(B_a))} \quad (3)$$

Because the annotated box only contains the vertebral body itself but not the very characteristic spinous processes, we decided to enlarge the box from which the features are extracted by a certain factor (cf. Fig. 5).

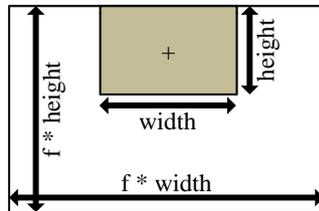


Figure 5. Illustration of the enlargement of the average annotation box (inner box) by a factor f in order to obtain the outer box being used for feature extraction. The location marked with the plus sign in the middle of the inner box marks the center of the enlargement process in the process of the candidate selection described in Sec. 2.2.

3.3 Results

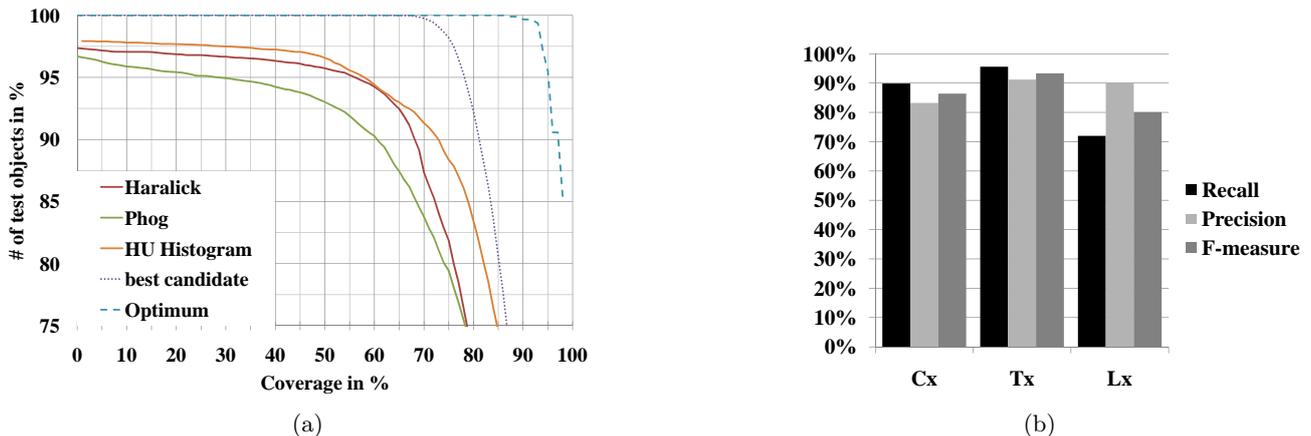


Figure 6. Comparison of the performance of different feature representations and the quality of the candidate selection (6(a)). And precision, recall and F-measure of the proposed algorithm using HU-histograms when classifying images into regions of the cervical- (Cx), thoracic- (Tx) and lumbar (Lx) spine (6(b)).

Feature Type Selection In the first set of experiments, we evaluated the suitability of the three feature representations (HU-Histogram, Haralick texture features, PHoGs edge features) for our detection method. Fig. 6(a) shows the results of these experiments. It can be seen that the orange line indicating the HU histogram features clearly outperforms the other feature descriptors as the histogram features always provide a higher overlap value than the other feature types in up to 60% of the cases. Exactly 60% overlap could be observed in 94% of the cases for both HU Histograms and Haralick features. However, the largest overlaps were only achieved by the HU Histograms (>88% overlap in 75% of the cases compared to 83% overlap in case of the Haralicks).

Candidate selection. Fig. 6(a) also shows the impact of the bone density map and the initial candidate selection. The dotted purple line indicates the performance that could be achieved if the candidate selection would always choose the correct candidate from the candidate set. The dashed blue line indicates a perfect match. Due to the fixed size of the box found by the algorithm and the different size of the true annotated box, there cannot be a 100% match in all cases.

The difference of the dotted best candidate line to the orange line of the histogram features indicates the error being caused by the process selecting the best candidate region. We observed that if the best candidate region would be selected in all cases, the coverage of 100% was achieved in up to 65% of the cases and a coverage of 87% was achieved in up to 75% of the cases. The difference between the dotted and the dashed line indicates the error which is caused by the generation of the candidate boxes.

Classification of the spine. Another experiment was conducted to check whether the HU Histograms would also be suitable to distinguish between cervical-, thoracic- and lumbar spine. Thus, we ordered the vertebral bodies into three classes (Cx: cervical-, Tx: thoracic-, Lx: lumbar spine) and classified each detected box into one of the three classes. To determine the class label of a new image, we selected the label of the closest neighbor of the result region in the sample set. The result of this experiment can be seen in Fig. 6(b) and Tab. 1. It can be seen that both precision and recall are greater than 70% for all three classes with the precision being greater than 90% in case of the thoracic- and lumbar spine and recall being greater than 90% in case of the cervical- and thoracic spine.

	Recall	Precision	F-measure
Cx	90%	83%	86%
Tx	96%	91%	93%
Lx	72%	90%	90%

Table 1.

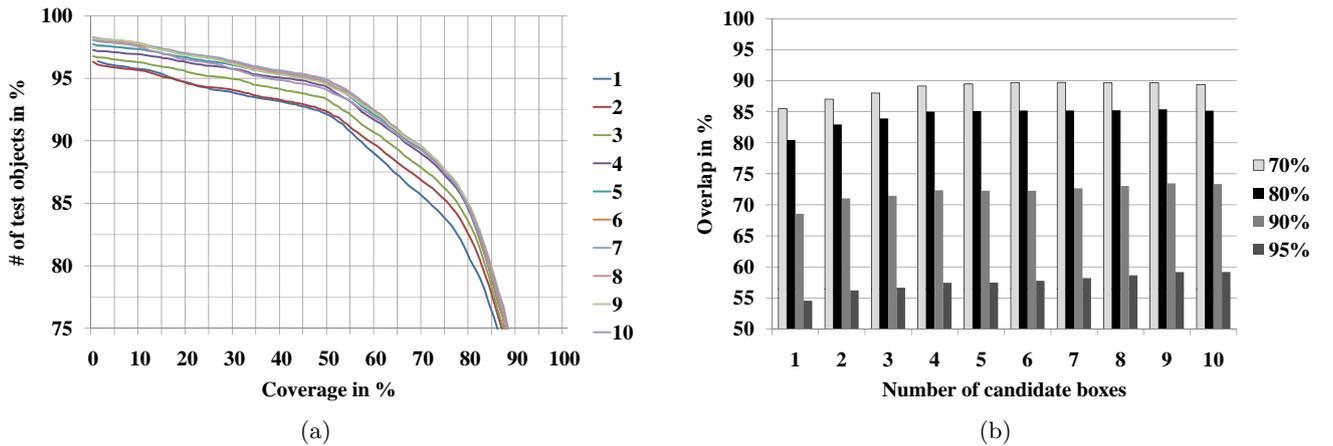


Figure 7. Impact of the number of candidate boxes (1-10) being selected from the bone density map on the classification performance. The bars marked as 70%, 80%, 90% and 95% indicate the amount of test cases achieving the given overlap.

Candidate boxes. As shown in the experiment above, the selection of the correct candidate box is very relevant for the performance of the algorithm. Thus, we evaluated the impact of the considered number of candidate regions (n_{cand}) to the performance of the algorithm. The result of this evaluation can be seen in Fig. 7. The fact that the lines in Fig. 7(a) are rather close to each other indicates that the amount of n_{cand} does not have a very large impact for $n_{cand} > 3$. The results displayed in Fig. 7(b) support this assumption as there is no significant difference in the performance for $n_{cand} > 4$. Thus, we conducted all further experiments with n_{cand} set to 5. The reason for choosing comparatively small number is that we have to perform a 1NN query in the sample set for each candidate region. Due to the large dimensionality of the employed feature descriptors, it is not possible to employ index structures to improve the runtime performance of nearest neighbor queries in the sample set. Therefore, a full table scan is required for each such query. As a result, increasing the number of considered candidate regions increases the search time linearly.

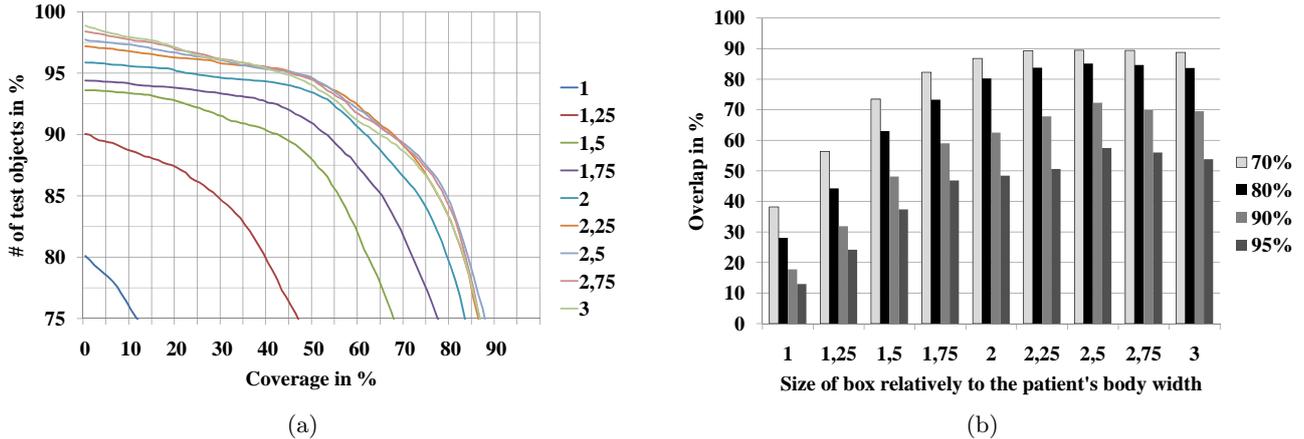


Figure 8. Impact of the size of the area from which features are extracted on the classification performance. The size of the box is regarded relatively to the average annotation box. The bars marked as 70%, 80%, 90% and 95% indicate the amount of test cases achieving the given overlap.

Size of candidate boxes. Another important factor having a large impact on the performance is the size of the box from which the image features are extracted. As mentioned before, the region from which the image features are extracted should be larger than the average annotation box because a region which only shows the vertebral body does not contain a lot of information. On the other hand, if the area is too large, the vertebrae will only cover a small part of the result region. Thus, we need to find a trade-off between accuracy and selectivity of the result region. To evaluate this trade-off, we enlarged the average annotation by a factor f between 1 and 3. The result can be seen in Fig. 8. The diagrams show a clear correlation between the performance of the algorithm and the enlargement of the annotation box with a slight maximum at about $f = 2.5$. The performance decrease for $f > 2.5$ is caused by the effect that the result region might be extended beyond the borders of the of the lower part of the image. Also the 3×3 grid from which the features are extracted is becoming rather rough in this case, making small important structures less significant.

Size of the inspected area of the image. Another source for errors is the bone density map. Especially if the resolution of the CT-scan along the z-axis is very high, it might happen that there is almost no bone structure in the area where the vertebrae is expected because the slice is exactly between two vertebrae and thus, only shows the intervertebral disk and the spinous process. In these situations, there is a comparatively high bone density along the ribs or at the sternum.

In Sec. 2.5, it is mentioned, that for performance reasons, it is possible to limit the search for bone pixels to the mid third of the detected image area. This step has not only the effect that the amount of pixel operations is being considerably reduced but also that the possible locations of the spine are limited to the most relevant area as well. Therefore, there exists another trade-off that must be dealt with: The algorithm could scan the

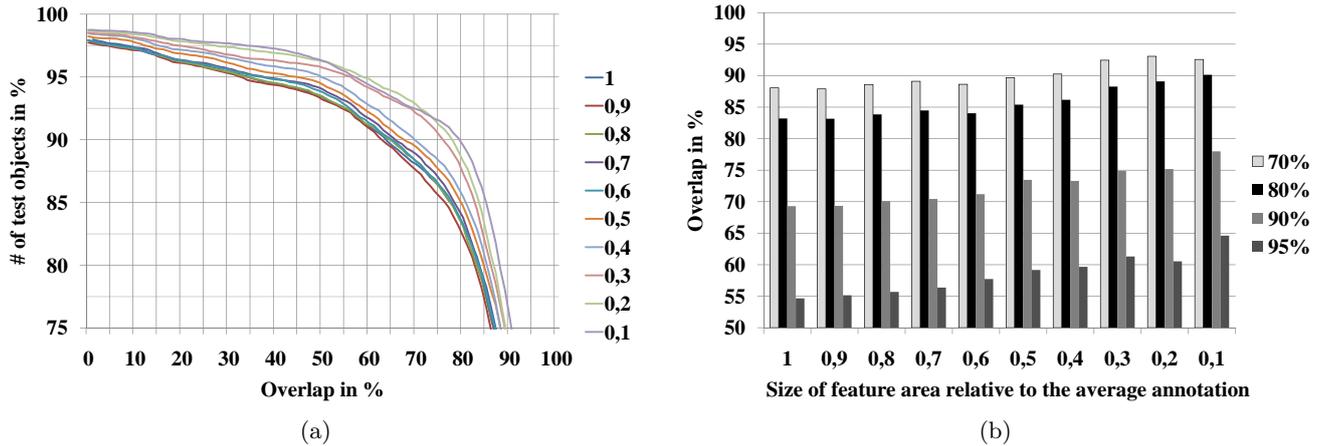


Figure 9. Impact of limiting the area being analyzed for bone pixels on the classification performance. The size of the analyzed area is regarded relatively to the patient’s body width. The bars marked as 70%, 80%, 90% and 95% indicate the amount of test cases achieving the given overlap.

complete image and thus provide highest flexibility at the cost of both runtime and accuracy. If the scanned area is very small, the processing speed is better but the classification might be less robust in cases where the vertebrae is outside the considered area. For example, if the bounding box detection fails, it might occur that the actual position of the spine is outside the considered area. Figure 9 shows the evaluation of this experiment and proves the assumption that the performance increases with a decreasing size of the scanned area. In order to choose a reasonable trade off between flexibility, processing time and accuracy, we choose a factor of $\frac{1}{3}$ for the evaluation of the algorithm.

Speed and Memory The processing time for all 9 239 images was 38 minutes on an AMD Athlon 2.59 GHz, which is about 240 ms per image in average in the case of HU Histograms (Haralick: 57 min, PHoG: 34 min). The settings used for this measurement comprised the complete process chain including the image preprocessing, bone density, candidate selection, nearest neighbor search (with the features kept in memory), refinement and validation. The algorithm was implemented in Java 1.6 with ImageJ.¹⁶ The memory footprint of the algorithm did not exceed the standard size of 32 mb for the heap of the Java VM.

4. CONCLUSION

In this paper, we proposed a new fast method for localizing the vertebrae on a 2D CT slice by using a novel and robust algorithm. Our method is based on a bone density map derived from the bone density and NN classification of texture features. To avoid a complete scan of the image, an iterative location refinement is employed. Our experiments indicate a coverage of the annotated ground truth of more than 83% in 80% of the cases. We evaluated different feature representations and compared the impact of several internal parameters to the performance of the algorithm. The test data is a large data base of more than 9 000 CT images. The major contribution of the presented work are the processing speed, a high accuracy, the robustness, the evaluation on a large test bed and that the user does not need to set any parameters by hand.

For future work, we will improve the detection quality of our proposed algorithm and further process the results of our detector to classify the detected vertebrae into the sub regions of cervical-, thoracic- and lumbar spine. Furthermore, we plan to improve the candidate detection and selection in order to improve the accuracy.

ACKNOWLEDGMENTS

This research has been supported in part by the THESEUS program in the CTC and Medico projects. They are funded by the German Federal Ministry of Economics and Technology under the grant number 01MQ07020. The responsibility for this publication lies with the authors.

We would like to thank Tomaz Vrtovec, Ph.D. (University of Ljubljana, Faculty of Electrical Engineering) for giving the initial idea for using the bone density as a first candidate selection.

REFERENCES

- [1] Vrtovec, T., Tomazevic, D., Likar, B., Travnik, L., and Pernus, F., “Automated construction of 3 D statistical shape models,” *Image Analysis & Stereology* **23**(2), 111–120 (2004).
- [2] Vrtovec, T., Likar, B., and Pernuš, F., “Quantitative analysis of spinal curvature in 3D: application to CT images of normal spine,” *Physics in medicine and biology* **53**, 1895 (2008).
- [3] Štern, D., Vrtovec, T., Pernuš, F., and Likar, B., “Automated determination of spinal centerline in CT and MR images,” in [*Proc. of SPIE Vol.*], **7259**, 72594M–1 (2009).
- [4] Štern, D., Likar, B., Pernuš, F., and Vrtovec, T., “Automated detection of spinal centrelines, vertebral bodies and intervertebral discs in CT and MR images of lumbar spine,” *Physics in Medicine and Biology* **55**, 247 (2010).
- [5] Z. Peng, J. Zhong, W. W. and h. Lee, J., “Automated vertebra detection and segmentation from the whole spineMR images,” in [*27th Int. Conf. of the Engineering in Medicine and Biology Society IEEE - EMBC '05*], (2005).
- [6] Klinder, T., Ostermann, J., Ehm, M., Franz, A., Kneser, R., and Lorenz, C., “Automated model-based vertebra detection, identification, and segmentation in ct images,” *Medical Image Analysis* **13**(3), 471 – 482 (2009).
- [7] Kim, Y. and Kim, D., “A fully automatic vertebra segmentation method using 3D deformable fences,” *Computerized medical imaging and graphics: the official journal of the Computerized Medical Imaging Society* (2009).
- [8] Nyúl, L., Kanyó, J., Máté, E., Makay, G., Balogh, E., Fidrich, M., and Kuba, A., “Method for automatically segmenting the spinal cord and canal from 3D CT images,” in [*Computer Analysis of Images and Patterns*], 456–463, Springer (2005).
- [9] Schmidt, S., Kappes, J., Bergtholdt, M., Pekar, V., Dries, S., Bystrov, D., and Schnörr, C., “Spine detection and labeling using a parts-based graphical model,” in [*IPMI'07: Proceedings of the 20th international conference on Information processing in medical imaging*], 122–133, Springer-Verlag, Berlin, Heidelberg (2007).
- [10] Corso, J., Alomari, R., and Chaudhary, V., “Lumbar disc localization and labeling with a probabilistic model on both pixel and object features,” *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2008* , 202–210 (2008).
- [11] Huang, S., Chu, Y., Lai, S., and Novak, C., “Learning-based vertebra detection and iterative normalized-cut segmentation for spinal MRI,” *IEEE transactions on medical imaging* **28**(10), 1595 (2009).
- [12] Haralick, R. M., Shanmugam, K., and Dinstein, I., “Textural features for image classification,” *IEEE Transactions on Speech and Audio Processing* **3**(6), 6103–623 (1973).
- [13] Emrich, T., Graf, F., Kriegel, H.-P., Schubert, M., Thoma, M., and Cavallaro, A., “CT slice localization via instance-based regression,” *Medical Imaging 2010: Image Processing* **7623**(1), 762320, SPIE (2010).
- [14] Canny, F. J., “A Computational Approach to Edge Detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence* **8**, 679–698 (1986).
- [15] Seifert, S., Kelm, M., Moeller, M., Mukherjee, S., Cavallaro, A., Huber, M., and Comaniciu, D., “Semantic annotation of medical images,” *Medical Imaging 2010: Advanced PACS-based Imaging Informatics and Therapeutic Applications* **7628**(1), 762808, SPIE (2010).
- [16] Rasband, W. S., “ImageJ,” U. S. National Institutes of Health, Bethesda, Maryland, USA, <http://rsb.info.nih.gov/ij/> (1997-2009).