Enhanced Detection of the Vertebrae in 2D CT-Images

Franz Graf[a] and Robert Greil[a] and Hans-Peter Kriegel[a] and Matthias Schubert[a] and Alexander Cavallaro[b]

[a]Institute for Informatics, Ludwig-Maximilians-Universität München,
Oettingenstr. 67, 80538 Munich, Germany
[b]Imaging Science Institute Erlangen, Maximiliansplatz 1, 91054 Erlangen, Germany

ABSTRACT

In recent years, a considerable amount of methods have been proposed for detecting and reconstructing the spine and the vertebrae from CT and MR scans. The results are either used for examining the vertebrae or serve as a preprocessing step for further detection and annotation tasks. In this paper, we propose a method for reliably detecting the position of the vertebrae on a single slice of a transversal body CT scan. Thus, our method is not restricted by the available portion of the 3D scan, but even suffices with a single 2D image. A further advantage of our method is that detection does not require adjusting parameters or direct user interaction. Technically, our method is based on an imaging pipeline comprising five steps: The input image is preprocessed. The relevant region of the image is extracted. Then, a set of candidate locations is selected based on bone density. In the next step, image features are extracted from the surrounding of the candidate locations and an instance-based learning approach is used for selecting the best candidate. Finally, a refinement step optimizes the best candidate region. Our proposed method is validated on a large diverse data set of more than 8 000 images and improves the accuracy in terms of area overlap and distance from the true position significantly compared to the only other method being proposed for this task so far.

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

1. INTRODUCTION

Detecting the position and the form of the spine or vertebrae is a useful task for various purposes in medical imaging. In spine reconstruction, information being extracted from a body scan is used to build a 3D model of the spine allowing the examination of spinal maladies like scoliosis. Further areas are not directly interested in the particular characteristics of the spine but rather employ the gained information to acquire positioning information within a scan. The position and the form of the spine are well-suited for determining the position of the patient on the table because vertebrae have different shapes in varying parts of the body. Furthermore, bone structures are well suited for automatic detection tasks due to their clear and invariant display in CT scans. To solve these problems the research community proposed several methods for detecting, segmenting, analyzing and reconstructing meshes of vertebrae and the complete spine being based on a complete 3D body scan. However, there are applications where a limited number of CT slices is available. For example, the slices being contained in a medical record which is sent to a specialist without access to the complete data in the PACS. To provide valuable information in this or similar applications, we focus on detecting the vertebrae in the smallest possible input, a single 2D image.

In this paper, we propose a new method for detecting the position of the vertebrae in 2D CT slices on the transverse plane. The information being derived by our approach can be employed for localizing the center of the body w.r.t. the position of the patient on the examination table. Additionally, our method is suitable for distinguishing the cervical-, thoracic- or lumbar spine area which yields value information about the body regions of the given image. Finally, our method shows reliable results even if larger regions of the considered body scan are available. Thus, it can be used as a preprocessing step for reconstructing 3D models of the spine as well.

To make our new method suitable for the named applications it meets several important requirements. For example, it is not necessary to adjust parameters to the characteristics of the given scan. Furthermore, the method

Further author information: graf@dbs.ifi.lmu.de, Telephone: +49 89 2180 9329 works fully automatic, i.e. the systems marks the spine in the image without any user interaction. Finally, our method works efficiently without requiring large amounts of main memory which makes it a suitable component for larger imaging systems.

Technically, our method employs five steps: In the first step, the input image is preprocessed by rescaling and smoothing. Then, the system extracts relevant pixels and weights each pixels. Both the search mask for determining the relevant pixels and the weighting functions are learned from an image repository containing slices with spine annotations. In the next step, we extract candidate locations in the selected area. Afterwards, image features for each candidate are extracted and the best candidate locations are selected for employing instance-based learning. In a final step, the annotation box around the best candidate location is fitted to the boundaries of the given vertebra.

The rest of the paper is organized as follows: Section 2 briefly surveys related work. The algorithm will be described in Section 3, followed by the description of our experimental setting and the achieved results in Section 4. Section 5 concludes the paper and gives an outlook to future work.

2. RELATED WORK

In the works of Vrtovec et al.[1–4] the authors construct 3D shape models of the spine and analyze the spinal curvature in CT images. Stern et al.[5,6] determine the spinal centerline in both CT and MR volume scans. There are also several methods aiming at the detection and segmentation of the spine.[7–9] Nyúl et al.[10] propose methods for detecting the spinal cord and the spinal canal in 3D CT scans by using deformable fences or models. Methods being based on 3D MR scans are proposed by Schmidt et al.,[11] Corso et al.[12] and Huang et al.[13] Though each of these methods are reported to achieve convincing results, all of them require a complete 3D volume scan. When considering this more challenging setting, the number of related methods is considerably smaller. Rangayyan et al.[14] use a Hough-Transformation to detect the spinal canal. However, the proposed method relies on reducing the search space to an area that should not be larger than a region comprising the vertebra. Graf et al.[15] propose a method for verterbrae detection from 2D slices which is the same task as in this paper. Therefore, we will compare to this method in our experimental evaluation and show how our new method outperforms this first solution to the given problem.

3. ALGORITHM

The detection of vertebrae consists of the five major steps, which are: image preprocessing, region extraction and weighting, candidate generation, candidate selection and refining the best candidate.

3.1 Preprocessing

In the first step, the original CT image I is rescaled to a unified width and height of 512×512 pixels. Afterwards, a 2D Gaussian kernel $G(x, y, \sigma)$ with $\sigma_{x,y} = 1$ is applied to I. The Gaussian blur has the effect of reducing noise in the CT slice. This noise can be caused by a high resolution along the z-axis of the body or in case of low-dose CTs.

3.2 Region Extraction and Weighting

On a transversal CT slice, the spine is always located in the lower middle region of the image if the scan was recorded in a dorsal position of the patient (which is the case in a huge majority of the cases). Thus, we limit the image I to the subregion which possibly could contain the spine by applying a search mask ρ_{sm}.

In contrast to the method of Graf et al.,[15] where the search space was set heuristically to $\frac{1}{3}$ of the patient's body, we determine ρ_{sm} empirically based on a training database denoted DB. As a result, we can further restrict the relevant region of the image without losing relevant information.

In particular, the training database DB consists of a set of CT volume scans $V_j \in DB$ where each volume scan is represented by an ordered set of images $I_{i,j} \in V_j$ $(i \in [1, n])$. Additionally, each vertebral body of a volume scan is annotated with a paraxial bounding box representing the ground truth. Thus, each $I_{i,j} \in V_j$ refers to a set $M_{i,j} = \{MBR(I_{i,j})\}$ that contains the minimum bounding rectangles (MBR) which are generated by the intersection of the annotation boxes of V_j with the image $I_{i,j}$. The cardinality of $M_{i,j}$ can thus be $[0..m]$ where m is usually no more than 3. $|M_{i,j}| = 0$ occurs, if the CT slice displays the section between two vertebral bodies, so that only a spinal disc is visible. $|M_{i,j}| = 1$ is the obvious case where the slice shows exactly one vertebral body. $|M_{i,j}| > 1$ occurs, if the rotation of the spine is large enough so that a slice along the transversal plane shows not only one vertebral body but also parts of neighboring vertebral bodies.

To define the search mask ρ_{sm}, we first create the union U of all sets of MBRs $M_{i,j}$ of all volumes $V_j \in DB$. Afterwards, we compute the convex hull of the set U using Graham's scan algorithm,[16] so that ρ_{sm} is defined by:

$$\rho_{sm} = ConvexHull(U) \tag{1}$$
$$U = \{\cup M_{i,j} \mid M_{i,j} = \cup MBR(I_{i,j} \in V_j) \wedge V_j \in DB\} \tag{2}$$

This step requires that DB is large and diverse enough. Otherwise, ρ_{sm} will be too selective and small so that no correct points of interest can be selected in the following steps. Building the convex hull around the set of ROIs is used to avoid that ρ_{sm} is overfitting to the database.

3.3 Candidate Generation

We aim to detect points within the image that are candidate locations for a vertebra. Compared to it's surrounding, a vertebra itself is a very locale bone structure. Thus, we create a bone density map for all suitable bone pixels $b_{x,y} \in \rho_{sm}$. Suitable bone pixels are all pixels $b_{x,y}$ with an HU-value in a certain HU window $[\beta_{lower}, \beta_{upper}]$. For our algorithm, we defined the window to the typical HU range for compact bones,[17] so that β_{lower} was set to 300 HU and β_{upper} was set to 1000 HU. Spongy bones which can be seen in an HU-range of $\beta_{lower} = 50 - 200$ HU can also be observed in the inner part of the vertebral bodies of elderly patients. Nevertheless, the outer part of the vertebral bodies is typically compact and thus in a higher HU-range, so that pixels with less than 200 HU need not be taken into account. Furthermore, a too low value for β_{lower} can lead to an increased rate of false candidates. The set B of suitable bone pixels is thus defined as

$$B = \{b_{x,y} \mid (x,y) \in \rho_{sm} \wedge p(x,y) \in [\beta_{lower}, \beta_{upper}]\} \tag{3}$$

where $p(x,y)$ denotes the HU-value at location (x,y). The bone density map D is then calculated by computing the accumulated distance for each bone pixel in the search mask to all other bone pixels within the search mask. The bone density $D(x,y)$ for a pixel (x,y) is then defined as

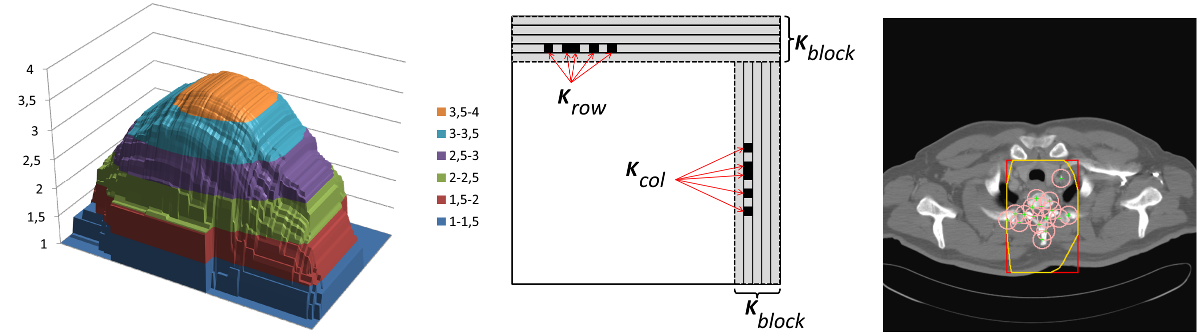
$$D(x,y) = \sum_{p \in B} (|x - p_x| + |y - p_y|) \tag{4}$$

with p_x and p_y denoting the x and y coordinates of a pixel p respectively. The L_1 distance is preferred to the L_2 distance due to faster computation. Thus, the bone pixels with the smallest accumulated distances $D(x,y)$ have the highest bone density because they are closer to other bone pixels than the pixels with larger accumulated distance values.

This rather simple distance map still leads to false candidate detections in the area of the sternum, the clavicle and the hips. In these cases, very dense bone structures extend into the search mask and might be selected as false candidate locations. A further refinement of the candidate region is not an applicable solution as the mask will either overfit or crop other true candidates from the search mask. For this reason, we propose to apply a weighting function w which is also derived from our training database DB:

$$w(x,y) = arg\ max(1,\ log(|R_{x,y}|)) \tag{5}$$
$$R_{x,y} = \{R_i \mid R_i \in ROI_{DB} \wedge (x,y) \in R_i\} \tag{6}$$

where ROI_{DB} is the set of all annotations in the database DB. Thus, $w(x,y)$ can be regarded as a measure for the likelihood that location (x,y) displays a true candidate. The logarithmic function is applied as a damping function in order to reduce the impact of pixels with large values of $|R_{x,y}|$ and thus to avoid overfitting. In general, any kind of monotone damping function with a co-domain of $[1, \infty[$ would be applicable. The weighting



(a) (b) (c)

Figure 1. (a): Illustration of the weighting function w. The height value denotes the value of $w(x, y)$. The bottom right of the illustration represents the bottom line of ρ_{sm}. Fig. (b) shows the parameters used in the refinement process. The column and row on the left/bottom are ommited for simplification. Fig. (c) shows the search mask ρ_{sm} (yellow line), and the densest pixels in $D(x, y)$ (green dots) including the λ neighborhood marked by circles.

function is then applied to the distance map D by a pixel wise multiplication (7) building the weighted distance map D_w. An illustration of w can also be seen in Fig. 1(a).

$$D_w = D \circ w \tag{7}$$

After the computation of the weighted bone density map D_w, the algorithm detects the η most promising candidate locations for the position of the vertebrae. This set will be denoted by C_{cand}. Simply extracting the η densest locations from D_w is not feasible because it is very likely that all of the η candidates are located within a small region around the global minimum of D_w. Thus, we first remove the densest pixel $(\{(x, y) \in D_w | \forall (u, w) \in D_w \ : D_w(u, w) > D_w(x, y)\})$ from D_w and add it as a first candidate to C_{cand}. This pixel is denoted by $c_{x,y}$. Afterwards, we remove all pixels from D_w in the λ-neighborhood of $c_{x,y}$ which is defined by:

$$N_\lambda = \{p_{x,y} | (x, y) \in D_w \wedge d(p_{x,y}, c_{x,y}) < \lambda\}; \tag{8}$$
$$d(p_{x,y}, q_{x,y}) = \sqrt{(p.x - q.x)^2 + (p.y - q.y)^2}. \tag{9}$$

This procedure is repeated until C_{cand} contains η candidates. The locations of the densest pixels is illustrated in Fig. 1(c).

3.4 Candidate Selection

After the extraction of the candidate locations, we extract image features from areas around each candidate location and select the most promising location based on an instance-based learner.

To extract the image features, we first center an ROI ϕ_{box} at each location in C_{cand}. The size of ϕ_{box} is set to twice the size of an average vertebral body. Thus, the texture information of the vertebral body includes the surrounding tissue. After positioning the ϕ_{box}, the image features are extracted and a feature vector $\phi_{desc} \in \mathbb{R}^n$ is built from the extracted data. As proposed in the works of Bosh et al.[18] and Graf et al.,[15,19] ϕ_{desc} is designed to take the spatial distribution of the pixel values into account so that shapes of objects in ϕ_{box} also have influence on the feature vector. This is done by sub-dividing ϕ_{box} in equally sized sub-regions from which the features are extracted. Afterwards, the features are serialized to build the feature vector $\phi_{desc(i)}$ with i denoting the type of feature.

In our experiments we evaluated different kinds of well known image features which are: HU-Histograms ($\phi_{desc(1)}$), Haralick texture features[20] ($\phi_{desc(2)}$), a derivate of Pyramid Histograms of oriented Gradients (PHoG)[18] ($\phi_{desc(3)}$), binary histograms ($\phi_{desc(4)}$) and resized image regions (thumbnails) ($\phi_{desc(5)}$). In particular, our image features are built by dividing the ϕ_{box} into $3 \times 3 = 9$ sub-regions and create a HU histogram for each sub-region.

The HU-histograms ($\phi_{desc(1)}$) consist of 16 bins corresponding to 16 equally sized intervals in the HU color space respecting the applied windowing ($[\beta_{lower}, \beta_{upper}]$). After building the histograms, all histograms are serialized in order to build a $16 \times 9 = 144$ dimensional feature vector $\phi_{desc(1)}$.

After building a ϕ_{desc} for each location in C_{cand}, we apply instance-based learning by determining the nearest neighbor of each $\phi_{desc} \in C_{cand}$ in the training database DB_{feat} containing the same feature description of the annotations in DB. For searching the nearest neighbor, we employ the L_1-distance function: $dist_{L_1}(\vec{u}, \vec{v}) = \sum_i |u_i - v_i|$. The location corresponding to the feature vector with the smallest distance to database annotation is finally selected as the best candidate position of the vertebrae.

3.5 Refinement

At this stage ϕ_{box} of the selected candidate describes the area which was used for feature extraction and thus, it is significantly larger than the vertebral body. This issue is addressed in this section, where the size of ϕ_{box} is reduced to fit the result box as tight as possible to the detected vertebra. Our dynamic refinement algorithm reduces the width and height of ϕ_{box} under consideration of the bone density at the borders of ϕ_{box}. A standardized scale factor is not applicable in this case, because the size of the vertebrae increases from the thoracic vertebrae to the last lumbar vertebrae. The refinement process is applied for each border (top, right, bottom, left) separately. In the following, we describe the method for relocating the top border downwards. The other borders are relocated respectively.

First, an ROI of κ_{block} pixel rows is aligned at the inner top row of ϕ_{box}. This ROI is moved downwards until each pixel row of the ROI contains at least κ_{row} pixels with an HU value greater than κ_{HU} or until the lower border of ϕ_{box} is reached. Thus, the y-location of the ROI can be defined as:

$$ROI.y = \{min(y) \in \phi_{box} | \forall row_i \in ROI : |p(x, y_i) > \kappa_{HU}| > \kappa_{row}\} \tag{10}$$

The resulting region is denoted by κ_{box}. If κ_{box} is empty (e.g. if the above ROI was moved up to the opposite side), the values κ_{block}, κ_{HU}, κ_{row} and κ_{col} respectively are softened (c.f. Tab. 1). If κ_{box} was refined too little, so that the area decrease is less than $\kappa_{\%}$ compared to ϕ_{box}, the values are hardened (c.f. Tab. 1) and the refinement is restarted in a second iteration. If the second iteration again produces a κ_{box} which is either too small or too large, κ_{box} is set back to the default size of ϕ_{box}. Finally, it is ensured that κ_{box} has a minimum size of $\kappa_{limit} \times \kappa_{limit}$ pixel which is achieved by rescaling the width/height accordingly if width or height fall below the limit. The default values used for the refinement process are shown in Tab. 1, illustrations of the ROIs are shown in Fig. 1(b).

κ_{block}	8 px	κ_{row}	8 px	κ_{col}	17 px
κ_{HU}	250 HU	$\kappa_{\%}$	0.04	κ_{limit}	60 px
κ_{block}	$+2/-2$ px	κ_{row}	$-3/+3$ px	κ_{col}	$-3/+3$ px
κ_{HU}	$-50/+100$ HU				

Table 1. Values for the refinement procedure. The upper 2 rows show the default values, the lower rows show the deltas for each value which are applied if the values are *soften*ed and *harden*ed respectively.

We also tested the approach proposed by Rangayyan et al. [14] to detect the spinal canal by using a Hough transformation and thus refine the position even further. Unfortunately the method was too unstable on our data so that no positive contribution of the method could be observed.

4. EXPERIMENTS

4.1 Data set

Our experiments are based on thorax CT scans of 34 different male and female patients (8 362 images). The scans were recorded with different CT scanners having various resolutions along the z-axis and some scans showing the application of contrast media. The scans also show differing body size, gender and age of the patients. The ground truth annotations were created by a clinician by annotating each vertebral body of the spine by an enclosing single 3D paraxial bounding box. The spinous processes were not included in the annotation boxes to avoid spatially large annotations and also large overlaps between consecutive vertebrae.

4.2 Quality Measure

Graf et al.[15] propose to employ the area overlap between the detected area and the annotation box which is doubled in height and width ($= \alpha_d$). Since their method does not use any refinement step, the result boxes have a fixed size. Thus, the size of the compared boxes is very similar. In our new method, the predicted region can be much smaller than α_d due to the refinement. Thus, a prediction being completely contained in the annotation box, can still be strongly displaced which indicates a bad prediction quality. To overcome this problem, we propose to employ a smaller annotation box. Thus, for our experiments the annotation box around the vertebral body is only extended vertically to the bottom by a factor of 2, building the new ROI α_v. Let us not that this extension is necessary to make sure that the spinal process is part of the annotation. Nonetheless, our new annotation boxes are only half of the size of the boxes employed in Graf et al.[15] Summarizing the overlap is measured as follows:

$$O_{former} = \frac{area(\alpha_d \bigcap ROI_{detected})}{min(area(\alpha_d), area(ROI_{detected}))} \tag{11}$$

$$\xi_{overlap} = \frac{area(\alpha_v \bigcap \kappa_{refined})}{min(area(\alpha_v), area(\kappa_{refined}))} \tag{12}$$

To have a second quality measure which is independent of box sizes, we also propose to measure the distance $\xi_{distance}$ between the centers of the ROIs α_v and $\kappa_{refined}$ in order to measure the spatial derivation of the search result to the true position.

$$\xi_{distance} = \sqrt{(p.x - q.x)^2 + (p.y - q.y)^2} \tag{13}$$

$$p = center(\alpha_v) \quad ; \quad q = center(\kappa_{refined}) \tag{14}$$

4.3 Evaluation

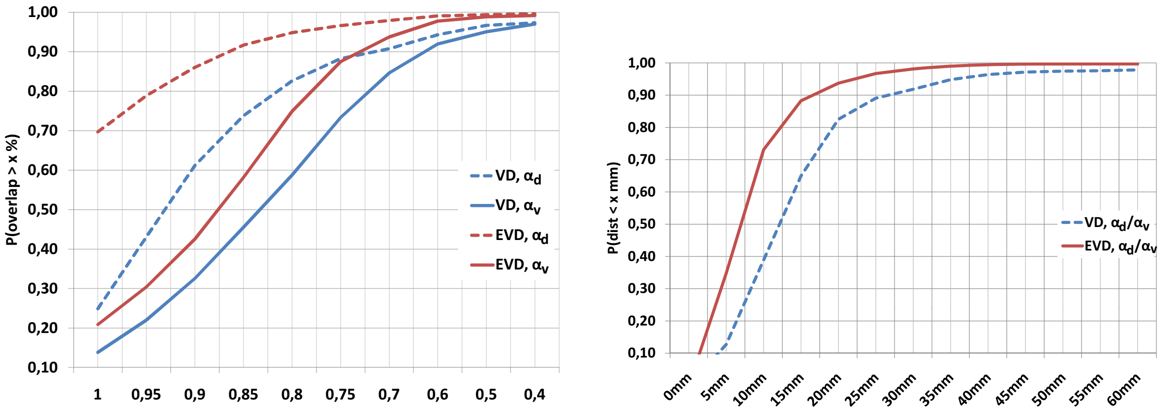
The evaluation was done by applying a cross validation, where a complete CT scan was defined as the source for query slices and all other scans were used as training data sets. This guarantees that slices from the same patient are not in the test and training set at the same time. In the following, we will compare our proposed algorithm (named **EVD**) to the state-of-the-art[15] named **VD**.

4.3.1 Comparison to VD and quality metric

Comparing EVD to VD shows a significant improvement throughout the complete cumulative distribution function (CDF) (Fig. 2(a)) measuring the overlap between the detected ROI and the annotated ground truth. Especially in the region of $70 - 80\%$ area overlap, an improvement of 15% can be observed. Also, $P(overlap > x\%) > 0.9$ is now reached at $x = 72\%$ of the cases compared to $x = 62\%$ using VD. Also the distance between the centers of the detected ROIs to the true ROI is reduced significantly as can be seen in Fig. 2(b): The detected ROIs now deviate less than 17 mm using EVD from the true position in 90% of the cases compared to 28 mm when using VD. As the former quality metric of simple area overlap is not very selective if the detected ROI becomes smaller, we show the performance of the algorithms in both quality metrics in Fig. 2. This figure also illustrates the impact of the new way the area overlap is calculated.

4.3.2 Impact of search mask ρ_{sm}

By applying the new region extraction, the search space in the image is reduced to an average of less than 9% of the original image's area. This is comparable to the simple approach proposed in Graf et al.,[15] where the search space was reduced to the mid third of the patients body, which caused an average reduction of less than 21% without employing any knowledge in the database.

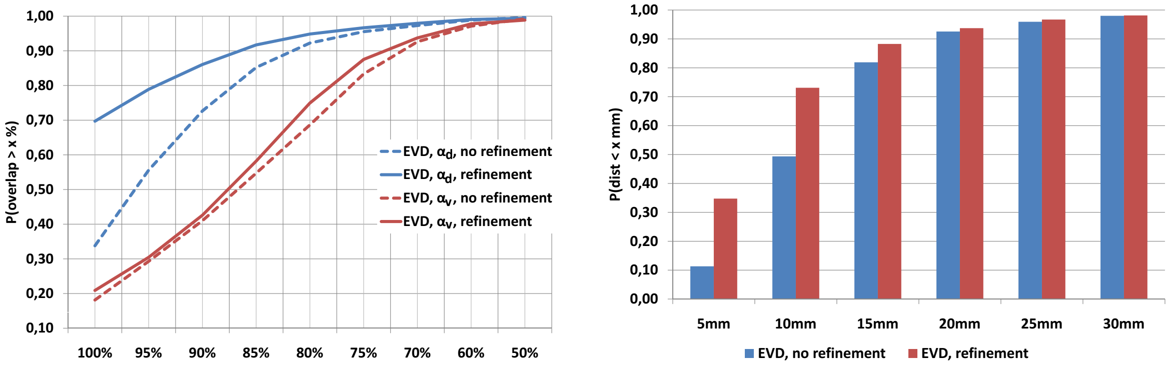


(a) (b)

Figure 2. Comparing VD and EVD by using the new overlap measure (a) and distance deviation (b). In Fig. (a) the difference of the dashed blue line to the solid blue line shows the impact of the stricter quality measure.

4.3.3 Refinement

One of the major concerns of VD was the size of the detected ROI which was defined by the size of an average annotation box doubled in width and height. The refinement method introduced in Sec. 3.5 addresses this issue and is able to reduce the width, height and area of the ROI to an average of 62% (width) and 80% (height) which results in an average area decrease of 49%. Also, the 2-step refinement fails in just less than 1% of the cases where the ROI is reset to the unmodified size of ϕ_{box}. Figure 3 shows the positive impact of the refinement process on both area overlap and distance deviation. Figure 3(b) shows the large impact on the distance deviation in the first two columns of the diagram. Using EVD, the probability to achieve $< 35\,\mathrm{mm}$ distance deviation is now 0.35 ($P(dist < 5\,mm) = 0.35$) compared to 0.11 using VD. Also $P(dist < 10\,mm)$ was raised from less than 0.5 to more than 0.72.



(a) (b)

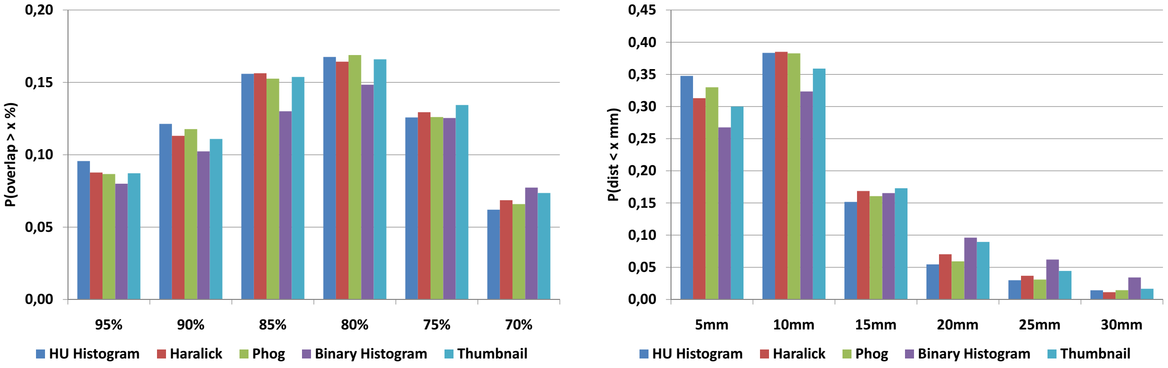
Figure 3. Impact of refining and of the tighter area overlap measured by area overlap (a) and by distance deviation (b). Fig. (a) also shows the impact on the area overlap using the new area overlap (red lines) and also the area overlap function using α_d as proposed for VD (blue lines).

4.3.4 Feature Descriptors / Candidate selection:

The evaluation of different feature descriptors ($\phi_{desc(1)}, \ldots, \phi_{desc(5)}$) mentioned in Section 3.4 showed that HU histograms perform better than the other tested feature descriptors. The diagrams shown in Figure 4 illustrate the results.

Regarding the area overlap (Fig. 4(a)), it can be seen that the different performance measures of HU histograms, Haralicks, PHoGs and Thumbnail features are quite close together up to $P(\text{overlap} > 0.80)$. Just the performance of the binary histogram features is clearly below the other features.

Regarding the distance deviation (Fig. 4(b)), HU histograms are more clearly outperforming the other descriptors in case of distance deviations of less than 5 mm (first column group). In case of distance deviations of less than 10 mm (second column group) HU histograms, Haralick and PHoG features perform almost equal. This diagram also demonstrates the needs for a second quality measure in order to obtain a more diverse view to the results.



(a) (b)

Figure 4. Comparison of feature types regarding both quality measures (Area overlap (a) and distance deviation (b)). Regarding both quality measures, HU histograms perform better than all other tested feature types.

Figure 5 shows the evaluation of the λ and η parameters which are both affecting the selection of candidate points (cf. Section 3.4). Obviously, the experiments proved a strong stability with respect to the values for both parameters so that neither the amount of candidates (λ), nor the size of the neighborhood η pose critical issues to the stability of the results.

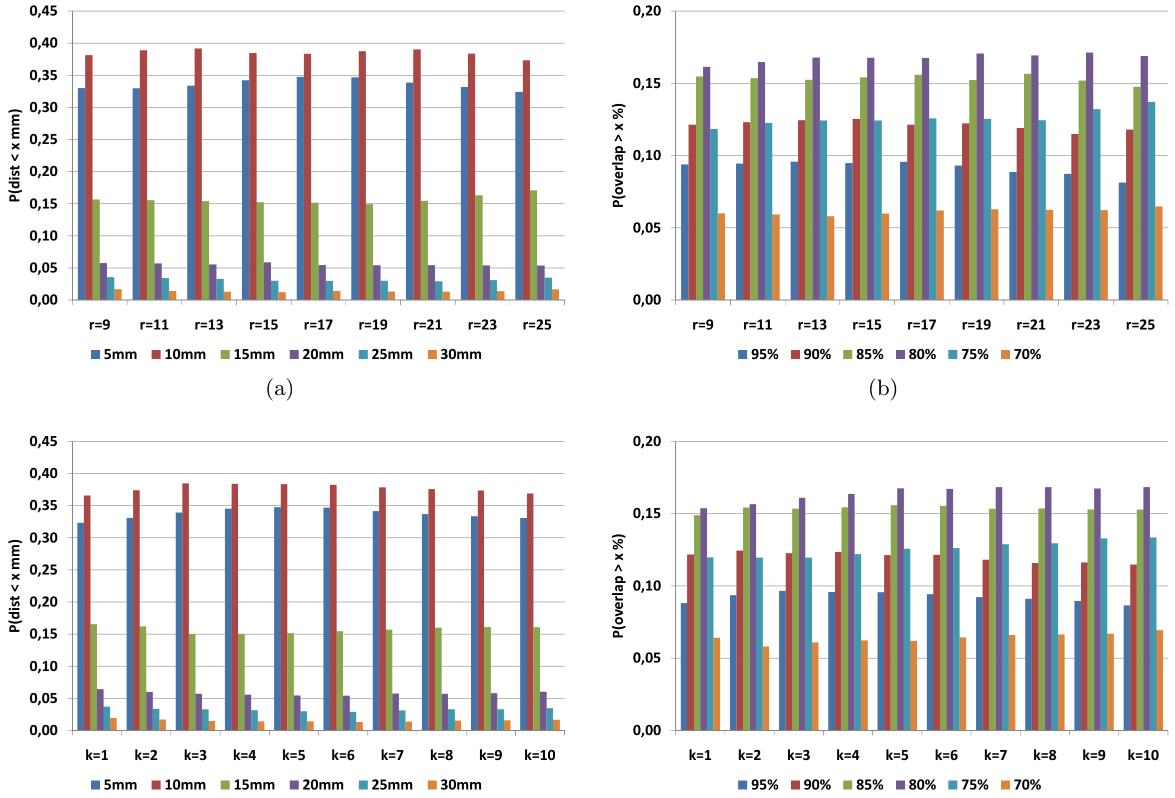
5. CONCLUSION

In this paper, we have proposed a new method for the automatic detection of vertebrae in 2D CT images. Our method is based on an imaging pipeline of five steps. In the first step, input images are rescaled and blurred. In the second step, we apply an empirically derived search mask and weighting function for the pixels within this mask. The third step, derives a weighted bone density map which is employed to select a set of candidate location. Step four of our pipeline selects the most promising candidate based on nearest neighbor selection on HU histograms of the candidate locations. In the final step, the selected candidate box is fit to the particular position of the vertebra. In our experimental evaluation, we compare our new method to a state-of-the-art method on a dataset being annotated by a clinical expert. The results indicate an increased prediction quality w.r.t. two quality measures, overlap and position derivation.

For future work, we plan to extend the work to 3D volume scans and also combine and compare the algorithm with segmentation and reconstruction. We also plan to test and add algorithms for metal artifact reduction[21] as artifacts can have significant impact on the algorithms.

Acknowledgements

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



(a) (b)

(c) (d)

Figure 5. Impact of the amount of candidates (η) ((a),(b)) and the size of the neighborhood λ ((c),(d)). It can be seen that the proposed method is rather stable in both parameters as no significant minimum can be observed.

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] Vrtovec, T., Pernus, F., and Likar, B., "Determination of vertebral pose in 3D by minimization of vertebral asymmetry," in [*Proceedings of the SPIE Medical Imaging 2011: Image Processing (SPIE), Lake Buena Vista, FL, USA*], (2011).

[4] Stern, D., Vrtovec, T., Pernus, F., and Likar, B., "Segmentation of vertebral bodies in ct and mr images based on 3d deterministic models," in [*Proceedings of the SPIE Medical Imaging 2011: Image Processing (SPIE), Lake Buena Vista, FL, USA*], (2011).

[5] Štern, D., Vrtovec, T., Pernuš, F., and Likar, B., "Automated determination of spinal centerline in CT and MR images," in [*Proceedings of the SPIE Medical Imaging 2009 Conference (SPIE), Lake Buena Vista, FL, USA*], **7259**, 72594M–1 (2009).

[6] 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).

[7] 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).

[8] 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).

[9] 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* **33** (2009).

[10] 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).

[11] 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).

[12] 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)* **5241**, 202–210 (2008).

[13] 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).

[14] Rangayyan, R. M., Deglint, H. J., and Boag, G. S., "Method for the automatic detection and segmentation of the spinal canal in computed tomographic images," *Journal of Electronic Imaging* **15**(3), 033007 (2006).

[15] Graf, F., Kriegel, H.-P., Schubert, M., Strukelj, M., and Cavallaro, A., "Fully automatic detection of the vertebrae in 2D CT images," in [*Proceedings of the SPIE Medical Imaging 2011: Image Processing (SPIE), Lake Buena Vista, FL, USA*], (2011).

[16] Graham, R. L., "An efficient algorithm for determining the convex hull of a finite planar set," *Information processing Letters* **1** (1972).

[17] Reball, S., [*Effekte der Physik und ihre Anwendungen.*], Deutsch Harri GmbH (2004).

[18] Bosch, A., Zisserman, A., and Munoz, X., "Representing shape with a spatial pyramid kernel," in [*Proceedings of the 6th ACM international conference on Image and video retrieval*], 401–408 (2007).

[19] Emrich, T., Graf, F., Kriegel, H.-P., Schubert, M., Thoma, M., and Cavallaro, A., "CT slice localization via instance-based regression," in [*Proceedings of the SPIE Medical Imaging 2010: Image Processing (SPIE), San Diego, CA, USA*], **7623**, 762320 (2010).

[20] 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).

[21] Golden, C., Mazin, S., Boas, F., Tye, G., Ghanouni, P., Gold, G., Sofilos, M., and Pelc, N., "A comparison of four algorithms for metal artifact reduction in CT imaging," in [*Proceedings of the SPIE Medical Imaging 2011: Image Processing (SPIE), Lake Buena Vista, FL, USA*], **7961**, 79612Y (2011).

[22] Graf, F., "JFeatureLib," (2011). http://code.google.com/p/jfeaturelib/.

[23] Felipe, J., Traina, A., and Traina Jr, C., "Retrieval by content of medical images using texture for tissue identification," in [*Computer-Based Medical Systems, 2003. Proceedings. 16th IEEE Symposium*], 175–180, IEEE (2003).

[24] Sivaramakrishna, R., Powell, K. A., Lieber, M. L., Chilcote, W. A., and Shekhar, R., "Texture analysis of lesions in breast ultrasound images," *Computerized Medical Imaging and Graphics* **26**(5), 303 – 307 (2002).

[25] Heimann, T., van Ginneken, B., Styner, M., Arzhaeva, Y., Aurich, V., Bauer, C., Beck, A., Becker, C., Beichel, R., Bekes, G., et al., "Comparison and evaluation of methods for liver segmentation from CT datasets," *Medical Imaging, IEEE Transactions on* **28**(8), 1251–1265 (2009).

[26] Canny, J., "A computational approach to edge detection," *Readings in computer vision: issues, problems, principles, and paradigms* **184**, 87–116 (1987).

[27] Deselaers, T., Weyand, T., Keysers, D., Macherey, W., and Ney, H., "Fire in imageclef 2005: Combining content-based image retrieval with textual information retrieval," in [*Accessing Multilingual Information Repositories*], Peters, C., Gey, F., Gonzalo, J., Müller, H., Jones, G., Kluck, M., Magnini, B., and de Rijke, M., eds., *Lecture Notes in Computer Science* **4022**, 652–661, Springer Berlin / Heidelberg (2006).

[28] Hurtut, T., Gousseau, Y., and Schmitt, F., "Adaptive image retrieval based on the spatial organization of colors," *Computer Vision and Image Understanding* **112**(2), 101 – 113 (2008).

APPENDIX A. PARAMETERS

	Annotation ROI
α_s	standard annotation box
α_d	α_s doubled in height and width
α_v	α_s doubled in height
	Feature Description/Extraction
$\phi_{desc(i)}$	Type of feature
ρ_{box}	ROI used for feature extraction
	Candidate selection
λ	neighborhood radius (pixels)
η	maximum amount of candidates
	Refinement
κ_{box}	result ROI after candidate selection
$\kappa_{refined}$	refined ROI
κ_{block}	minimum of rows/colums for borders
κ_{row}	minimum number of pixels per row
κ_{col}	minimum number of pixels per column
κ_{HU}	lower bound for pixel detection (HU)
$\kappa_{\%}$	minimum area difference to κ_{box} (%)
κ_{limit}	minimum width/height of final ROI
	Quality measure
$\xi_{overlap}$	area overlap
$\xi_{distance}$	distance

Table 2. List of all parameters used in our algorithm including their meaning and measures.

APPENDIX B. ADDITIONAL FEATURE DESCRIPTORS

Most feature descriptors used in this work were obtained or derived from features implemented in JFeatureLib.[22]

B.1 Haralick Texture Features

Haralick texture features[20] are not uncommon in medical image processing, PACS retrieval[23] or tissue detection.[24,25] These features describe the spatial correlations in gray level images between adjacent pixels by calculating statistics based on the co-occurrence matrix of the pixel values. In the original work, 13 different yet strongly correlated features are proposed which are created under consideration of a certain distance. Again, we divide ϕ_{box} into $3 \times 3 = 9$ sub-regions and compute for each sub-region all image features for five different distance values $(1, 3, 5, 7, 11)$. The resulting features are concatenated and build a feature vector $\phi_{desc(2)}$ of $9 \times 13 \times 5 = 585$ dimensions.

B.2 PHOG (Pyramid Histograms of Oriented Gradients)

In the past years, gradient representations have gained significant interest in the imaging community. Thus we apply a slightly modified PHOG descriptor.[18] In the original work, histograms of gradient orientations were extracted from a region of interest (ROI), afterwards this ROI is subdivided in 4 equally sized sub regions and the process is started recursively on each of these sub-regions until a specified maximum level is reached. The recursive sub-division builds a pyramid of ROIs from which the histograms are extracted. For edge detection, we are using the well known Canny edge detector[26] and extract the first two levels of the pyramid. In contrast to Bosch et al.[18] we decide to split the ROI into 3×3 sub-regions instead of 2×2. This is done because we extract features from ϕ_{box} which is significantly smaller than the complete image. Another reason is the trade off

Name	Sub-regions	dimensionality	Short description
HU-Histograms	9	144	HU-range
Haralick	9	585	Texture features
PHoGs	9	120	Edge detection
Binary-Histograms	121	121	binarized HU-scale
Thumbnails	none	256	HU value

Table 3. Overview of the tested feature descriptors and their dimensionality.

between spatial conformity and dimensionality of the resulting feature vector. For each sub-region, we compute a 12 bin gradient histogram and build a single feature vector $\phi_{desc(3)}$ consisting of $(9 + 1) \times 12 = 120$ dimensions.

B.3 Binary-Histograms

Binary histograms are a modified version of HU-histograms. In the binary case, the value of a histogram bin i describes the amount of pixels $(x, y) \in ROI_i$ that have an HU value greater than a threshold τ so that

$$bin_i = \{|(x,y)| : (x,y) \in ROI_i \land p(x,y) > \tau\} \tag{15}$$

with $p(x, y)$ denoting the HU value of a pixel at location (x, y). Thus, each bin of a histogram denotes the amount of pixels $> \tau$ of an ROI_i. As an ROI_i only generates a single value, we can divide ϕ_{box} into more sub regions without creating a feature vector with a too high dimensionality. In our case divide ϕ_{box} in 11×11 sub ROIs an use $\tau = 300\,\text{HU}$ which can be regarded as a lower bound of HU values for compact bones [17, p. 233]. Thus we create a feature vector $\phi_{desc(4)} \in \mathbb{R}^{121}$ representing the distribution of bone pixels $\in \phi_{box}$.

B.4 Thumbnails

Same as color histograms, down sampled images are also often used as a base line in image retrieval tasks.[27, 28] In this case, the image to be analyzed is rescaled to a uniform size to $x \times y$ pixels where w and h are usually less than 32 pixels. The resulting $w \cdot h = d$ pixels are then for example converted to gray scale and used as the features of a feature vector $fv \in \mathbb{R}^d$. In our case we do not rescale the complete image but only the according ROI. chose $w = h = 16$. The HU values of the resulting 256 pixels are used to build the feature vector $\phi_{desc(5)} \in \mathbb{R}^{256}$.

APPENDIX C. REFINEMENT ALGORITHM

Algorithm 1 Pseudocode for the refinement process. The delta of the values applied in the *soften/hardenParameters()* are displayed in Table 1 of the main part of the paper.

```
$\kappa_{refined} \leftarrow refine(\kappa_{box})$
if $isEmpty(\kappa_{refined})$ then
    $softenParameters()$
    $\kappa_{refined} = refine(\kappa_{box})$
else if $area(\kappa_{refined}) > \kappa_{\%} \cdot area(\kappa_{box})$ then
    $hardenParameters()$
    $\kappa_{refined} = refine(\kappa_{box})$
end if
$\kappa_{refined}.height \leftarrow$ increase if $width < \kappa_{limit}$
$\kappa_{refined}.width \leftarrow$ increase if $height < \kappa_{limit}$
$result \leftarrow \kappa_{refined}$
```