Fast scale-invariant lateral lumbar vertebrae detection and segmentation in X-ray images

Ruhan Sa[*], William Owens [†], Raymond Wiegand[¶] and Vipin Chaudhary[*]
[*]Department of Computer Science and Engineering
State University of New York (SUNY) at Buffalo, Buffalo, NY 14260–2500
[†]School of Medicine and Biomedical Sciences
State University of New York (SUNY) at Buffalo, Buffalo, NY 14214-3013
[¶]Spine Metrics, Inc. 520 Huber Park Court Weldon Spring. MO 63304

***Abstract*—Fully automatic localization of lumbar vertebrae from clinical X-ray images is very challenging due to the variation of X-ray quality, scale, contrast, number of visible vertebrae, etc. To overcome these challenges, we present a novel framework, where we accelerate a scale-invariant object detection method using Support Vector Machines (SVM) trained on Histogram of Oriented Gradients (HOG) features and segmenting a fine vertebra contour using Gradient Vector Flow (GVF) based snake model. Support Vector Machines trained on HOG features are now an object detection standard in many perception fields and have demonstrated good performance on medical images as well. However, the computational complexity and lack of robustness brought by rescaling the original images have prevented its applicability. The proposed multi-stage detection framework uses lower-level detection result to determine the rescaling regions to reduce the region of interest, thereby decreasing the execution time. We further refine the detection result by segmenting the contour of vertebra using GVF snake, where we use edge detection techniques to increase the robustness of the GVF snake. Finally, we experimentally demonstrate the effectiveness of this framework using a large set of clinical X-ray images.**

I. Introduction

Although spinal sagittal morphology has been defined for quite some time in medical field, it was only until recently that both researchers and clinicians have put forth more effort to understand and measure newly defined parameters. When specialists consider the automation of the Spine Measurement Software system (SMS), they consider vertebrae detection and localization technology as a way to produce a detailed 3-D model of the spine with all the appropriate measurements required to make efficient and accurate decisions on patient care. However, due to the variations in terms of machine settings, patient location, patient posture, X-ray intensities, number of visible vertebrae, etc., in X-ray images, full automation of this task can be very challenging. Fig. 1 shows some of the challenging variations that exist in our dataset.

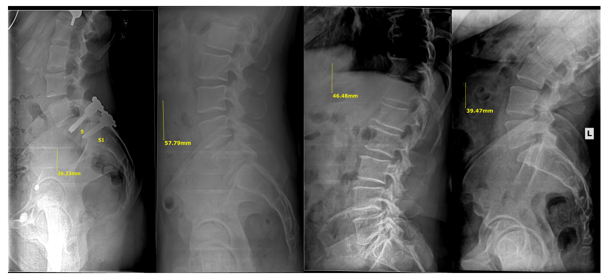

Fig. 1. Variations of lateral lumbar X-ray images. From these images we can see that the intensities varies considerably; the curvature of the spine differs; image contrasts is low; and the number of visible vertebrae differs. These factors are some of the challenges for fully automatic vertebrae detection and segmentation.

Spine image analysis has been studied extensively on different medical image modalities. There are a number of methods proposed for MRI images [1], [3], [4]. However, algorithms used for MRI images can hardly migrate to X-ray images, due to the loss of information on coronal and axial slices and low contrast. In recent studies, a few X-ray based methods have been proposed. Eduardo *et al* [6] proposed a semi-automatic method, where center points of vertebrae are localized manually followed by a neural network on Gabor magnitude response to classify the pixels into vertebra pixels and non-vertebra pixels. Dong *et al* [7] also proposed a semi-automatic method, where they first manually locate the first and last visible vertebrae and based on which they utilized a graphical model to further analyze the image. Moura *et al* [5] proposed an intensity based method. However, intensities vary a lot in our dataset (as they are from different x-ray machines) and would not be a robust feature. Fabian *et al* [2] present a fully automatic vertebra detection method using edge polygonal approximation and SVM trained on SIFT description. However, their edge polygonal approximation depends on the Canny edge detector, which results in unstable performance due to contrast differences between images. SIFT descriptor is also unstable for the same reason. In this paper, we demonstrate a novel framework, where we use a multi-stage SVM classifier on HOG features to robustly detect the vertebrae, acquire the bounding boxes, and use GVF snake to further detect and refine the contour of the vertebrae based on dynamically refined Canny edge detection result and hough filter result. Our method is significantly faster than existing methods. Object detection has been a widely studied area in computer vision and Linear SVM trained on HOG features has become popular due to its performance and robust theory [8]. However, the computational complexity due to rescaling the original image has been a major drawback. In medical image analysis, this drawback is accentuated by its demand for high accuracy. Moreover, the visible vertebrae number varies from image to image so that the size of the bounding box cannot be determined or assumed beforehand. Most of the X-ray images have more than 3,000,000 pixels. Thus, having a finely rescaled image pyramid can be computationally expensive and infeasible. To overcome this issue, we propose a multi-stage detection method where we take advantage of object classification result to reduce the rescale region significantly. Another contribution of our framework is that instead of using GVF snake models on original image, we improved the robustness of the method by detecting the edge within the bounding box first and perform GVF snake deformation on the edge detection results. Canny edge detection has been used widely in edge detection application, however, selection of gaussian kernel can affect the signal-to-noise ratio considerably. Thus, we dynamically choose the gaussian kernel in our framework to eliminate the bias and use hough filter to further process the edge to decrease the noise level. The method is described in detail in the following section.

II. Vertebrae detection and segmentation

A. Multi-stage vertebrae detection

Fig. 2 and 3 outline our proposed multi-stage framework. The first stage of our framework is to use parts of the object as the target, *i.e.,* train SVM classifier on HOG features of object parts. SVM trained on HOG features are now an object detection standard in many perception fields and one of the key advantage of HOG feature is that it is invarient to geometric and photometric transformation. In this step, we use object parts as target. Since the original image scale is unpredictable, we assume that object parts are easier to detect than the whole object. We iterate the first stage until we find object parts. In our experiments, object parts can almost always be found in the first iteration without rescaling the image. When using multiple iterations, we rescaled the image to a smaller size in every iteration until we found object parts. At this point, we will have a set of small parts represented by rectangular boxes. Let's say it is $P = \{p_1, p_2, ..., p_n\}$, where P is the set of the parts and n represent total number of object parts. Each $p_i, (i = 1, 2, ..., n)$ can be represented by two diagonal corner points, say $((x_{i1}, y_{i1}), (x_{i2}, y_{i2}))$, where x and y represent the (x, y) coordinates. We denote the iteration number in the first stage as itr_1. At the end of the first stage, we compute the large bounding box containing all parts in P, where we set the maximum and minimum (x, y) coordinates as the ROI, which we represent as $ROI = ((x_{min}, y_{min}), (x_{max}, y_{max}))$. From here, we will focus on ROI and the rescaling in next stage will be done on ROI instead of the entire image.

The second stage of the framework rescale the ROI to find the entire object. Due to the largely reduced region, the time spend on this stage can decrease substantially. Let's say S is the image size, which is typically above 2000x2000 pixels and we can represent the size of ROI as:

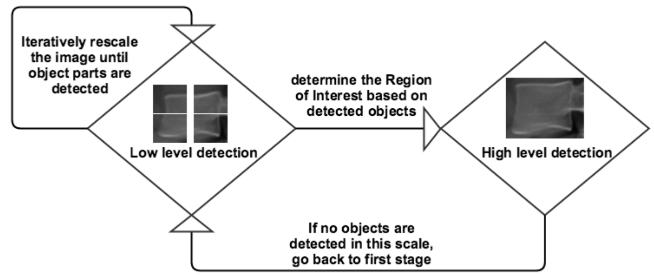

Fig. 2. Work-flow of our fast detection algorithm. First stage (on the left) iterate until object parts are detected and use the bounding box of the result as ROI. Second stage (on the right) uses full object detection.

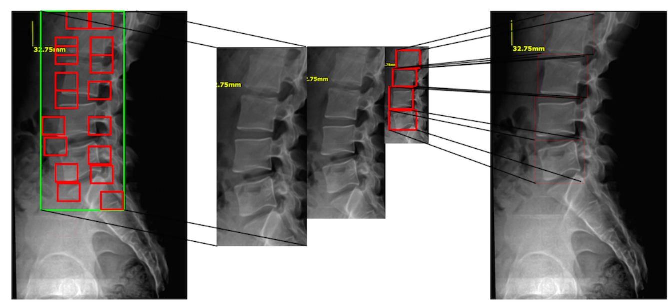

Fig. 3. Multi-stage detection. Image on the left illustrates the first stage of detection, using object parts as samples. Middle image illustrates how the rescaling has reduced to ROI, thus reducing classification time; and the image on the right shows the detection result added back to the original scale.

$$S_{ROI} = S - (x_{max} - x_{min})(y_{max} - y_{min}) = cS, \quad (1)$$

where $c \in [0, 1]$ is a constant, typically around 0.5 in our case. The time complexity of the entire framework can be represented by

$$T = \sum_{i=1}^{itr_1} T(\alpha^{i-1}S) + \sum_{i=itr_1+1}^{itr_1+itr_2} T(\alpha^{i-1}cS), \quad (2)$$

where $T(s)$ is a detection time on an image with size s; itr_2 is the iteration number in stage two; α is rescaling factor. However, in conventional method the time taken is:

$$T = \sum_{i=1}^{itr_1} T(\alpha^{i-1}S) + \sum_{i=itr_1+1}^{itr_1+itr_2} T(\alpha^{i-1}S) \quad (3)$$

Subtracting the above equation 1 from 3, we will have the saved time, which is

$$t = \sum_{i=itr_1+1}^{itr_1+itr_2} [T(\alpha^{i-1}S) - T(\alpha^{i-1}cS)] \quad (4)$$

Since each image pixel is computed multiple times, let's assume $T(s) \propto s^n$, where $(n > 1)$. Then above equation will be

$$t = \sum_{i=itr_1+1}^{itr_1+itr_2} [\alpha^{i-1}S]^n(1 - c^n) \quad (5)$$

We can see that with the increase of n, *i.e.* when the reuse time of a pixel in HOG feature extraction phase increase or with the decrease of c, *i.e.* the region of interest is small enough, the execution time decreases.

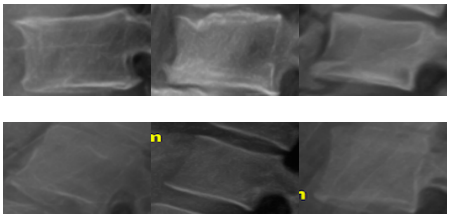

Fig. 4. Variations of L2 vertebra. We can see that although it is the same vertebra structure, it appears very different. Some vertebrae are tilted to the right and some are flat.

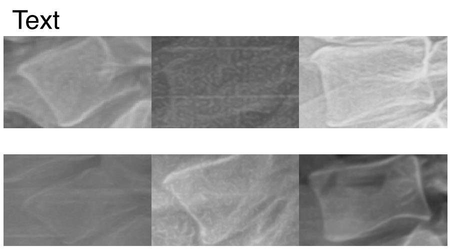

Fig. 5. Variations of L5 vertebra. Similar to the previous figure, L5 vertebra appears to be very different for distinct patients.

Apart from the fast framework, inspired by exemplar-SVMs proposed by Malisiewicz [9], we categorize the vertebra into three classes: right-tilted, flat and left-tilted classes instead of using different labels for each vertebra or using one label for all vertebrae. The reason is that the classification is based on spine structure, for example one classifier for each L1,L2,..,L5 vertebra can decrease the detection accuracy as the appearance of the same vertebra can vary a lot due to different patient posture, image scale, etc. Furthermore, HOG feature does not have sufficient support for rotation of rigid objects, which makes it hard to correctly describe different states of the same vertebra. Fig. 4 and Fig. 5 show examples of the same vertebra appearing very differently on different images. One classifier for all vertebrae is also inadequate and not robust, as we know that each vertebra appears differently, even in one image. Hence, we classify objects into three classes based on their rotations.

B. Vertebra segmentation

In this part, we utilize the bounding box obtained from the first phase to extract fine contour of vertebrae. GVF has been used widely due to its good performance in snake deformation. Alomari *et al* [3] demonstrated good performance of GVF based vertebrae disc segmentation by using joint model. The difference is that since X-ray images are not as clear as MRI images, we need to do further preprocessing. We first detect the vertebrae edge using Canny edge detection and hough filter. Since Canny edge detection is very sensitive to the contrast of the image, we implemented a dynamic gaussian kernel size determination method to deal with various X-ray quality issues, where we increase the

Fig. 6. Workflow of processing image within bounding box. Image on the left shows the initial Canny edge detection result and the middle image shows the final Canny edge detection result after dynamically chosen gaussian kernel size and the image on the right shows the hough filter result, which is used in the following GVF snake.

Fig. 7. Final GVF snake result.

kernel size until the foreground and background ratio hits a certain threshold. Thus, we can make sure that the resulting image will have as little noise as possible. After the edge detection, we use hough filter to further eliminate some minor noises. Fig. 6 shows the process of this step.

After we obtain a relatively nice contour, we use GVF snake model to build the complete close contour. One advantage of snake deformation on GVF in our example is that most of the time, because of the blurriness of the image, it is not easy to get a close contour, which makes the other methods such as using hough filter alone, insufficient. As shown in Fig 6, the contour is not closed circle. If we use other methods, such as hough filter to detect horizontal and vertical lines, we will not be able to produce a close contour, and hough filter also gives false positives, such as the horizontal lines shown on the middle right side of the image. Thus, we use GVF snake deformation to deal with missing edges. Fig. 7 shows the GVF snake results for entire lumbar vertebrae.

TABLE I
DETECTION RATE

dataset number	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
True positive	4	2	3	3	6	2	2	7	7	4	5	3	3	4	5	5	3	4
Total number of vertebrae	5	5	4	5	6	4	4	7	7	5	5	5	5	4	5	5	5	5

TABLE II
EXECUTION TIME COMPARISON.

	our method	conventional method
Avg time (min) per image	3.1	5.4
Avg iteration number	2.1	2.3

III. EXPERIMENTAL RESULTS AND ANALYSIS

A. Data and environment

To conduct our experiment, we used 30 clinical lateral lumbar X-ray images with varying parameters, in terms of number of visible vertebrae, intensities, noise level, etc. In order to accommodate the scale invariant feature of our framework, we analyzed the vertebrae size of all our samples, and chose the smallest size as our HOG window size, which is 200x300 pixels. For HOG settings, we used 8 orientations, 16x16 cell size and 1x1 block size. Linear kernel is used for SVM as suggested in [8]. Lower level SVM classifier is trained on 4127 positive samples and 72452 negative samples, with 200x300 image size. Higher level SVM classifier is trained on 175 flat, 247 right-tilted, 175 left-tilted, 22563 negative samples. GVF iteration number is 80, snake parameters are set as $dmax = 20, dmin = 5$ for snake interpolation, $\alpha = 0.05, \beta = 0, \gamma = 1, \kappa = 0.5, iteration = 35$, and $step - size = 5$ for snake deformation. The experiment is conducted on a 2GHz Intel Core i7 machine with 8GB, 1600 MHz DDR3 machine on serialized program.

B. Detection rate

We demonstrate the quality of our detection method based on true positives numbers versus total visible vertebrae number. Table I shows the performance of each individual test data. The average true positive rate achieved was 75%.

C. Execution Time

As shown theoretically in Section II-A, we further compared the time consumption of our detection framework and conventional detection framework. We trained on 20 clinical images and tested on 18 images with an average of 5 vertebrae, 4,342,388 pixels. Table II shows the average time consumption of our method versus conventional method without accelerated framework . From the data we see that, even though the average number of iterations are similar, the average execution time is reduced significantly. Average iteration number differs because the region of interest is changing for our method, thus the alignment issue comes into affect.

IV. CONCLUSION

In this paper, we proposed a novel and fast scale-invariant framework for detecting and segmenting vertebrae from X-ray images. This framework uses multi-stage detection to reduce the rescaling region for HOG feature based SVM classifier and we theoretically and experimentally demonstrated the time saved by using this new framework without compromising the accuracy of the classifier. Our framework easily handles different scaling issues in X-ray images. This framework also utilized an exemplar-SVMs like classifier to better represent the vertebrae. In the segmentation phase, we implemented dynamic selection of gaussian kernel in Canny edge detection phase, which ensures the signal-to-noise ratio and we further reduced the noise by using hough filter. Using edge detection results has given GVF snake better performance, especially when the X-ray images have poor contrast. Even though we have largely speeded up the algorithm, we still have room for improving the execution time. As our future work, we will be focusing on algorithmically improving the speed of framework as well as utilizing parallel computing techniques to improve the performance. We will further improve the robustness of this method by experimenting with more data.

REFERENCES

[1] Schmidt, S., et al. *Spine detection and labeling using a parts-based graphical model*, In Information Processing in Medical Imaging, 2007 (pp. 122-133). Springer Berlin/Heidelberg.

[2] Lecron, Fabian, et al. *Fully automatic vertebra detection in x-ray images based on multi-class SVM.*, SPIE Medical Imaging. International Society for Optics and Photonics, 2012.

[3] Alomari, Raja S., et al. *Lumbar spine disc herniation diagnosis with a joint shape model.*, Computational Methods and Clinical Applications for Spine Imaging. Springer International Publishing, 2014. 87-98.

[4] Ghosh, Subarna, et al. *A supervised approach towards segmentation of clinical MRI for automatic lumbar diagnosis.*, Computational Methods and Clinical Applications for Spine Imaging. Springer International Publishing, 2014. 185-195.

[5] Moura, Daniel C., et al. *Automatic vertebra detection in x-ray images.* CompIMAGE. 2006.

[6] Ribeiro, Eduardo, et al. *Segmentation of vertebrae in lateral lumbar spinal X-ray images* SP-ASC, 2012.

[7] Dong, X. and Zheng, G., *Automated Vertebra Identification from X-Ray Images,*, in Image Analysis and Recognition, Lecture Notes in Computer Science 6112, 19 (2010)

[8] Bristow, Hilton and Lucey, Simon, *Why do linear SVMs trained on HOG features perform so well?*, in arXiv preprint arXiv:1406.2419 (2014).

[9] Malisiewicz, Tomasz, et al. *Ensemble of exemplar-svms for object detection and beyond.*, Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, 2011.