

Automatically Extract Vertebra and Compute the Cobb Angle based on Spine's Features and Adaptive ASMs in Posteroanterior Radiographs

Pham The Bao

Computer Science Department, Information Science Faculty
Sai Gon University, Hochiminh city, Vietnam

Abstract—Nowadays, clinical diagnoses are more and more supported by medical equipment, but doctors still need a lot of time and effort to diagnosis. The construction of a system that can diagnose automatically will help doctors a lot when the doctor must handle many medical records. Many diseases need to be diagnosed by radiographs and they can be diagnosed automatically, especially bone diseases. The purpose of this paper is to introduce a new method to measure the curve of the spine in the X-ray images. We split this subject into two problems. The first problem is to extract the spine from the X-ray image. We use the threshold to remove redundant information from the images. Then an automatic mask is created to save the position of the spine and smooth the boundary of the spine. Based on the spine extracted from the previous steps, the Active Shape Model (ASM) which is designed from the characteristics of the vertebrae, is used to extract each vertebra from the spine. Finally, we measure the Cobb angle which is formed by vertebrae. The solution of two problems is implemented on X-ray images which have high quality, the results are more than 80% an area of the spine is extracted, and the Cobb angle is measured correctly, the accuracy of our method will decrease if the quality of the image is low.

Keywords—Spine detection; spine extraction; vertebrae detection; cobb angle; adaptive active shape model

I. INTRODUCTION

A spine is a series of vertebrae extending from the skull to the small of the back, enclosing the spinal cord and providing support for the thorax and abdomen [Oxford Dictionaries]. Its responsibility is to support the body, move, and protect spinal marrow [1]. In the past and even in the present, spinal deformation is almost depended entirely on observing and diagnosing of orthopedic surgeons. Nowadays, there is a lot of model equipment like X-ray machine which can support doctors to diagnose diseases of the spine, especially scoliosis. To recognize scoliosis, doctors measure the Cobb angle, and this angle is defined as the angle formed between a line drawn parallel to the superior endplate of one vertebra above the fracture and a line drawn parallel to the inferior endplate of the vertebrae one level below the fracture. If the Cobb angle is more than 10 degrees, the spine is considered to be scoliosis. When society is more and more modern, many people want to take care of your health. It leads to a large number of medical records with many tasks such as reviewing, measuring, analyzing, and classifying; a doctor can not undertake all of

them. The automatic system with artificial intelligence, which can detect vertebrae and calculate the Cobb angle from an X-ray image, will help doctors a lot. This system also helps other applications to model spine from 2D to 3D and detect diseases relating to the spine such as osteoporosis, gout in the spine, scoliosis, spondylolisthesis.

Many solutions have been applied to automatically detect lung, kidney, and bone from the radiograph. The aim of detecting the lung method [2, 3] is to identify the common features of the lung in an X-ray image and then use classifications. In another way, active models (Active Appearance Model - AAM [4] and Active Shape Model - ASM [5]) and their corresponding statistics [6] have been applied with the same basic idea, both of them try to fit the model of the sample (obtained by training data) to a new pattern by searching on the surrounding of points of new pattern. Smyth [7, 8] used the quantitative morphometry (QM) to build ASM of the lumbar vertebrae in the Dual-energy X-ray absorptiometry (DEXA) image to measure the shape. Robert [9] used AAM, which is characterized by a parameter model of the object. Finally, a method which gets good result is proposed by Luc Duong [10]. He reduces the noisy spine radiographs by using anisotropic filtering, and then the filtering process and pattern classification are devised to detect the spinal curve.

In this paper, we propose a new approach based on the characteristics of the X-ray image and the spine. This method is presented in five steps. First, we identify the region of interesting (ROI) in the X-ray images, which includes the spine [11], and the dynamic threshold which bases on the features of those areas, are applied for the ROI. Then we record the position of the spine in the image by an automatic mask. Third, based on the position of the spine, we proceed to adjust the spine correctly; this method is easier to apply than other methods, which is presented above. Fourth, we apply ASM to extract each vertebra from the spine. Finally, the Cobb angle is measured, and our system will give comments about whether the patients have scoliosis. Moreover, if scoliosis appears more than once in a spine, for example, Fig. 9, there are two signals of scoliosis (a left top curve and a right bottom curve), our method can solve that problem which other methods do not mention about.

II. MATERIALS AND METHODS

A. General System

The input X-ray images must be taken posteroanterior and include the spine from the neck to the hipbone. The spine also has to be in a normal upright position, as shown in Fig. 1.

Our method is started by using thresholds to remove unnecessary pixels and follows as the flowchart 1.

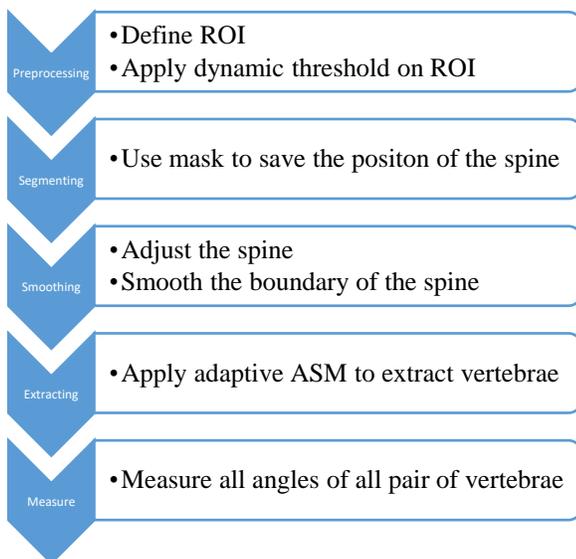
B. Preprocessing

First of all, the X-ray image is grayed out because of its pixels that just change the bright between black and white.

A normal posteroanterior radiograph of the spine embraces two parts: upper half of the image has less bright than the lower, but the upper has more contrast than the lower. It's because the lower half of the image is the region that contains many X-ray resistant things, organs [1]. Thus, we split this image into two parts vertically and create thresholds for each part base on its features.



Fig. 1. Input Image.



Flowchart. 1. Steps to Solve the Problem.

Each part of an X-ray image of the spine has a different threshold. We analyze the number of spinal pixels with its brightness's value statistically. Throughout empiric statistics for feedback characteristic X-ray of the spine, we have formula 1, where ave is the average brightness of the pixels having nonzero value of the image, max is the highest brightness's value of pixels of an image, α is coefficient of variation, and $0 < \alpha < 1$.

$$threshold = ave + (max - ave) \times \alpha \quad (1)$$

The threshold will only change if only α changes. Moreover, the threshold of the upper half of the image is lower than that of the lower half. With knowledge from doctors and experiments, we had found that α should be approximately $1/3$ to get a good result for the upper half of the image. We remove all the pixels which are darker than the threshold in each part. We have a result in Fig. 2.

C. Segmenting

We start segmenting by binarizing an image. Almost people's bodies always are in an upright position, except sleeping time. Moreover, the middle spine is the region that changes status so much; for example, the middle spine must hunch when a person stops or turns the body. Hence, the middle part of spine is the most flexible part and the most pressure of the spine [11]. So, if the spine is scoliosis, the middle part is the most affected part. In addition, the size of vertebrae increases from the cervical region to the lumbar region [1]. We used morphology method to remove some small objects (noises). From these characteristics, we will detect the middle spine first.

Considering a horizon line in middle binary images, we refer the longest line segments with the white pixels passing through areas that are the backbone. The white pixels in the image are not continuous and fragment into sections, those lead to difficulties in finding the longest line segment. To handle this situation, we solve this problem by connecting all pixels in an error span (t) to a continuous segment line. We design the algorithm 1 – detecting the middle horizon line of spine.

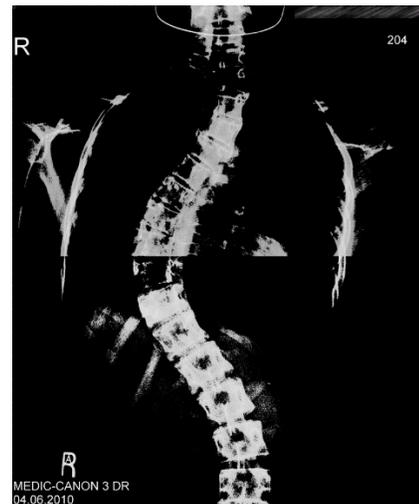


Fig. 2. Result of the Preprocessing.

```

Algorithm 1
Input: A binary spine image,  $A = \{a_{i,j} / i = \overline{1..r}, j = \overline{1..c}\}$ .
Output: middle line of spine,
 $L_{mid} = \{(x_{lmid}, y_{lmid}), (x_{rmid}, y_{rmid})\}$ .
j = 1;
mid = [r/2];
end = begin = lj = j;
while (j < c) do
  if ((end - left) < (j - lj)) then {
    begin = lj;
    end = j;
  }
  if (A(mid,j) is black) then j++;
  else {
    lj = j;
    flag = 0;
    while (j < c and !flag) do
      if (A(mid,j+t) is white) then j=j +t;
      else {
        end = j;
        begin = lj;
        flag = 1;
      }
    endwhile
  }
endwhile
xrmid = xlmid = mid;
yrmid = end;
ylmid = begin;

```

```

Algorithm 2
Input:  $L_{mid}$  – result of algorithm 1. Defining value of neighbor of point (x,y) on x-axis, it was called
 $NE(x, y) = \{(x, y \pm \Delta y) / \text{value of point is white}\}$ .
Output: spinal region,  $P_{left} = \{(x_l, y_l)\}$  and  $P_{right} = \{(x_r, y_r)\}$ 
x = xlmid = xrmid;
 $P_{left} = P_{right} = \emptyset$ ;
 $P_{left} = P_{left} \cup \{(x_{lmid}, y_{lmid})\}$ ;
 $P_{right} = P_{right} \cup \{(x_{rmid}, y_{rmid})\}$ ;
yl = ylmid;
yr = yrmid;
while(x still have value) do
  x = x  $\otimes$  1; // operator  $\otimes$  is “+” in lower-spine process and “-” in upper-spine process.
  if(NE(x,y1) or NE(x,yr) is not empty) then
    {
      yl = getCorrect(NE(x,y1));
      yr = getCorrect(NE(x,yr));
       $P_{left} = P_{left} \cup \{(x, y_l)\}$ ;
       $P_{right} = P_{right} \cup \{(x, y_r)\}$ ;
    }
  endwhile
// function getCorrect(NE(x,y)) return the y-correct on spine in the set NE(x,y) based on value of pixel //and direction of y-axis of left and right boundary of spine.

```

We use this segment line to determine the spine by moving it – L_{mid} – to the upper row and lower row from the middle of the spine. The determination process is executed as algorithm 2 – determining 02 parts of the spine; it was called upper-spine (UP) and lower-spine (LP). $UP = \{P_{left_up} \oplus P_{right_up}\}$, $LP = \{P_{left_low} \oplus P_{right_low}\}$ with $P_k = \{(x_i, y_i)\}$. We have resulted in Fig. 3.

D. Smoothing Boundary

After the segmentation step, the spine case is shown clearly, and we remove all the pixels that are outside the spine case by algorithm 3, and we have resulted in Fig. 4.

Although the obtained spinal that case contains most of the spine, the border is folded much. This problem leads to the deduction of some bones. We conduct a smoothing boundary algorithm to solve this problem, algorithm 4. The result is in Fig. 5.

Now, we have the new boundary of the spine. Most of the information on the spine is inside this boundary. With a normal spine or scoliosis without any disease else, its right or left boundary is a continuous curve that does not change the direction immediately.



Fig. 3. Result of the Region of the Middle Spine.

```

Algorithm 3
Input: UP and LP
Output: SP – the spine of image and CL – center line of
spine.  $SP = \{P_{left} \oplus P_{right}\}$ ,  $CL = \{(x_i, y_i)\}$ .
 $SP.P_{left} = UP.P_{left} \cup LP.P_{left}$ ;
 $SP.P_{right} = UP.P_{right} \cup LP.P_{right}$ ;
Sorting  $SP.P_{left}$  by  $x$  increase.
Sorting  $SP.P_{right}$  by  $x$  increase.
 $CL = \emptyset$ ;
for each  $x \in SP.P_{left}$  do
    for  $y = 0$  to  $c$  do
        if
            ( $y < SP(x).P_{left}.y_l$  or  $y > SP(x).P_{right}.y_r$ ) then set  $A(x,y)$ 
            is black
        endif
         $CL = CL \cup \{(x, (SP(x).P_{left}.y_l + SP(x).P_{right}.y_r)/$ 
        2)\};
    endif
endif

```



Fig. 4. Result of Segmentation.

We model and predict the curve by a regression model. This model uses the weighted linear least squares and the second-degree polynomial model. Then we use the robust version of this model by assigning lower weight to outliers in the regression. The boundary which is modified is shown in Fig. 6. After we have a new boundary of the spine, we update the CL.

E. Extracting Vertebrae

The region of the spine was extracted correctly; we perform to detect each vertebra in this region. The structure of any normal vertebrae is the same as each other when we view it from the same direction [13]. Hence, we propose to use Active Shape Models (ASMs) to detect vertebrae. We obtain a statistical model by marking 30 points on each vertebra and

using ASM to train them. We use the statistical model of the vertebrae to locate better positions for points. For each landmark point, the Mahalanobis distance is used to relocate its position for a better result in our method.

However, if we look into the size, it is much difference between any pair of vertebrae. Moreover, with scoliosis, a vertebra can have many directions (a vertical position or a left tilt position or a right tilt position) [13, 14]. With the difference in the size and direction of the vertebrae, if we just use one sample model for all vertebrae, the result will not good. Thus, we design many sample models to detect vertebrae.

```

Algorithm 4
Input: SP – the spine image. SP was sorted by x-axis. Value
d, the ratio of segment.  $\alpha$  – The angle of x-axis and built line
segment.
Output: SP – the updated spine image.
//smoothing for right boundary of spine.
for each  $x \in SP.P_{right}$  form the smallest to biggest do
     $t_x = (SP(x).P_{right}.y_r - SP(x).P_{left}.y_l)/d$ ;
     $I_x = \emptyset$ ;
     $\beta = \alpha$ ;
    while (  $I_x$  is empty) do
        Building the line segment  $l_x$  at point
 $A_x(x, SP(x).P_{right}.y_r)$  with angle  $\beta$  and the length
        is  $t_x$ . The form of line segment is
 $l_x(t) = \beta t + \gamma$ .
         $I_x = l_x \cap SP.P_{right}$ ;
        Increase  $\beta$ ;
    endwhile
    for  $k = x$  to  $I_x.x$  do
         $SP(k).P_{right}.y_r = l_x(k)$  // update
        value of right boundary of spine
    endfor
endfor
//smoothing for left boundary of spine.
for each  $x \in SP.P_{left}$  form the smallest to biggest do
     $t_x = (SP(x).P_{right}.y_r - SP(x).P_{left}.y_l)/d$ ;
     $I_x = \emptyset$ ;
     $\beta = \alpha$ ;
    while (  $I_x$  is empty) do
        Building the line segment  $l_x$  at point
 $A_x(x, SP(x).P_{left}.y_l)$  with angle  $\beta$  and the length
        is  $t_x$ . The form of line segment is
 $l_x(t) = \beta t + \gamma$ .
         $I_x = l_x \cap SP.P_{left}$ ;
        Decrease  $\beta$ ;
    endwhile
    for  $k = x$  to  $I_x.x$  do
         $SP(k).P_{left}.y_l = l_x(k)$  // update
        value of left boundary of spine
    endfor
endfor

```

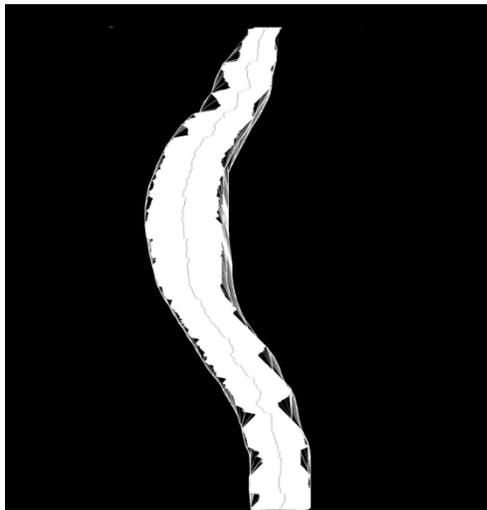
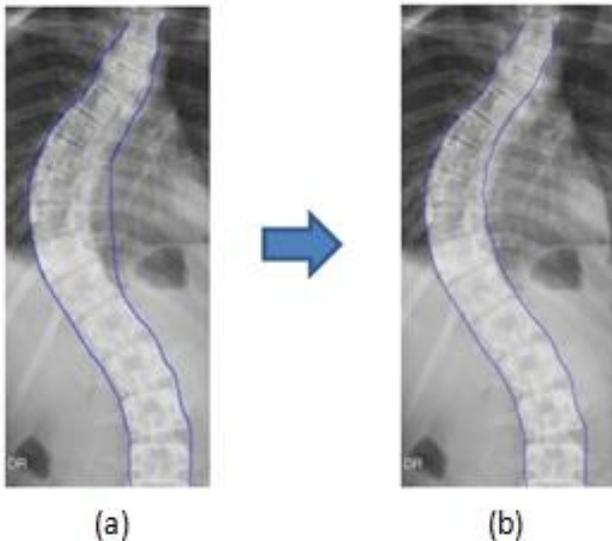


Fig. 5. A New Boundary of the Spine.



(a) Before using the Regression Model. (b) After using the Regression Model.

Fig. 6. The Boundary of the Spine.

At the bottom region of the spine, the vertebrae are much larger than the one in the top region [14], one model of ASM cannot change the size in a large range between the top vertebrae's size and bottom vertebrae's size. Thus, we have two models; one is a large model for the bottom region, and one is a small model for the top region. However, the height of vertebrae is increased steadily from top to bottom, Table I [13, 14].

The main purpose of this paper is to detect the spine, whether it is scoliosis or not, and the Cobb angle is the measurement to detect this. Simplify, The Cobb angle is defined as the angle formed by the intersection of two lines, one parallel to the superior endplate of a vertebra above the fracture and another line parallel to the inferior endplate of the vertebrae one level below the fracture. If the Cobb angle is more than 10 degrees, that is the symptom of scoliosis. The Cobb angle may be more than 60 degrees.

TABLE. I. VERTEBRAE INFORMATION

Kinds of vertebrae s	Averaged size (cm)	Number of vertebrae s
Cervical	12.5	7
Thoracic	28	12
Lumbar	18	5
Sacral	12.5	5
Coccyx		4

TABLE. II. LIST OF ASM MODELS

ASM model	Size	Direction
ASM_1	Large	Upright
ASM_2	Large	Left
ASM_3	Large	Right
ASM_4	Small	Upright
ASM_5	Small	Left
ASM_6	Small	Right

To cover all cases of vertebrae in scoliosis, we design three kinds of models of ASMs for each small model and large model above, each of three models which can rotate within at least 20 degrees angle. In other word, a model can rotate at least 10 degrees to the left and 10 degrees to the right. Specifically, first kind of model – normal model – is designed in the same direction of the vertical axis of spine, it can rotate 10 degrees to the left or the right; the second – left model – is designed to have the direction which slopes 20 degrees to the left of the spine's vertical axis; the third – right model – is the same of the second, but it slopes to the right. These ASM models, which are used, are developed by Tim Cootes and Chris Taylor [12]. In a word, with the different size and direction of each vertebra, we design six sample models as Table II. One model is trained by at least four samples which have the same corresponding characteristics.

To detect each vertebra more correct, we process three steps. Firstly, we detect all vertebrae s from bottom to top – algorithm 5 – we have $V_{up} = \{V_{up(i)}\}$ and the second is top to bottom – algorithm 6 – we have $V_{down} = \{V_{down(i)}\}$. The last, we check the correlation from results V_{up} and V_{down} ; because the middle region of the spine is complicated to have the set of vertebrae $\{V_i\}$. The accuracy of bottom-up detection is higher than top-down detection because the last vertebra is stable and very clear to be detected. We use μ – the angle of the line (we regress all points on the centerline of the spine) with y-axis, the direction of vertebrae and the position of vertebrae to detect kind of ASM model, Table III and formula (1). We always detect the last and the first vertebrae of the spine, then the remaining sequential vertebrae. The intervertebral disc is between two vertebrae (V_{i-1} and V_i). We must calculate the size of the intervertebral disc ($inter_{disc}$), which is dependent on the height of two vertebrae [13, 14]. If V_{i-1} and V_i are the same feature then by ASM model of V_{i-1} and ASM model of V_i together in one of three sets: $\{AMS_1, AMS_4\}$, $\{AMS_2, AMS_5\}$ and $\{AMS_3, AMS_6\}$. Normally, the height of the intervertebral disc is approximately 20% of the height of the vertebrae [13, 14]. We use formula (3) to compute the height of the intervertebral disc. We have resulted in Fig. 7 and 8.

$$typeASM(x_1, x_2, \mu, L_{mid}) = \left[\sqrt[4]{\frac{\mu}{10}} \times [sign(x_1, x_2) + 2] + [sign(x_1, x_2, L_{mid} \cdot x) + 1] \right] \quad (2)$$

with $0 \leq \mu \leq 150$, $x_1 > 0$, $x_2 > 0$ and $L_{mid} \cdot x > 0$

$$sign(x_1, x_2) = \begin{cases} 0 & \text{if } x_1 > x_2 \\ 1 & \text{otherwise} \end{cases} \quad \text{and } sign(x_1, x_2, L_{mid} \cdot x)$$

$$= \begin{cases} 0 & \text{if } L_{mid} \cdot x > \frac{x_1 + x_2}{2} \\ 1 & \text{otherwise} \end{cases}$$

$$inter_{disc} = \left(\frac{d}{100} \right) \times V_{i-1} \cdot height \quad (3)$$

with d

$$= \begin{cases} 20 & \text{if } V_i \text{ and } V_{i-1} \text{ is the same in feature} \\ sign(V_i \cdot centerline, V_{i-1} \cdot centerline) & \text{otherwise} \end{cases}$$

$$sign(V_i \cdot centerline, V_{i-1} \cdot centerline)$$

$$= \begin{cases} 0 & \text{if } a > 20 \\ (20 - a) & \text{otherwise} \end{cases}$$

a is degree angle between center line of

V_i and center line of V_{i-1}

TABLE. III. APPLIED MODELS FOR VERTEBRAE

μ	Direction of vertebrae	Position of vertebrae	Kind of model
$\mu < 10$ degrees		Lower half of spine	ASM_1
$\mu \geq 10$ degrees	$x_1 > x_2$	Lower half of spine	ASM_2
	$x_1 < x_2$	Lower half of spine	ASM_3
$\mu < 10$ degrees		Upper half of spine	ASM_4
$\mu \geq 10$ degrees	$x_1 < x_2$	Upper half of spine	ASM_5
	$x_1 > x_2$	Upper half of spine	ASM_6

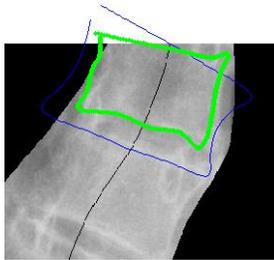
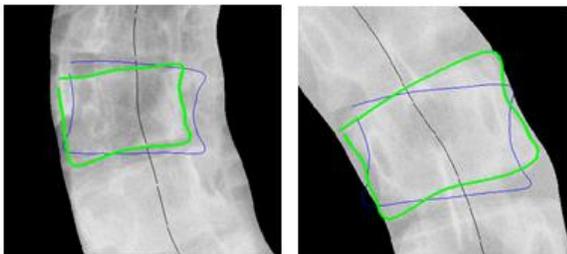


Fig. 7. Apply ASM for the Top Vertebrae: the Sample Model is in Blue; the Result is in Green Color.



(a) Middle vertebrae

(b) Bottom vertebrae.

Fig. 8. Apply ASM for the Sample Model is in Blue Color; The Result is in Red Color.

Algorithm 5. – **detecting all vertebrae s from bottom to up.**
Input: SP, CL, L_{mid} and $ASM = \{ASM_i / i = 1..6\}$
Output: $V_{up} = \{V_{up(i)}\}$ – the set of vertebrae s, which are located from down to up of spine.
// detect the last vertebrae of spine
 $h = \frac{\sum_{i=1}^3 ASM_i \cdot height}{3}$;
// check value of h with table 1.
 $x_1 = \max(CL \cdot x)$;
// x_2, x_1 are on CL
 $x_2 = x_1 - h$;
 $temp = \frac{(x_1 + x_2)}{2}$;
 $\mu = \text{getAngle}(x_1, x_2, CL)$;
 $type = \text{getTypeOfASM}(x_1, x_2, \mu, temp, L_{mid})$;
//type = 1..6
 $ASM_{type} \cdot center = (temp, CL \cdot y_{temp})$;
 $ASM_{type} \cdot window = \begin{pmatrix} x_1, x_2, SP \cdot P_{left}(\text{from } x_1 \text{ to } x_2), \\ SP \cdot P_{right}(\text{from } x_1 \text{ to } x_2) \end{pmatrix}$);
 $ASM_{type} \cdot active$;
 $V_{up(1)} = \text{result of } ASM_{type}$;
update CL, SP, x_2 from ASM_{type} ;
// detect the remaining vertebrae
 $i = 2$;
 $type_{old} = type$;
while (data remains on CL) do
 $h = V_{up(i-1)} \cdot height$; **// the height of previous located vertebrae was changed after ASM was run.**
// the height of vertebrae is decreased by the result from ASM models.
 $x_1 = x_2$;
 $x_2 = x_1 - h$;
 $temp = (x_1 + x_2) / 2$;
 $\mu = \text{getAngle}(x_1, x_2, CL)$;
 $type = \text{getTypeOfASM}(x_1, x_2, \mu, temp, L_{mid})$;
//type = 1..6
 $inter_{disc} =$
 $\text{getInterVertebralDisc} \left(\begin{matrix} V_{up(i-1)}, \\ type, type_{old}, x_1, x_2 \end{matrix} \right)$; **// the size of intervertebrae l**
//disc
if($inter_{disc} > 0$) then
 $x_1 = x_1 - inter_{disc}$;
 $x_2 = x_1 - h$;
 $temp = (x_1 + x_2) / 2$;
endif
 $ASM_{type} \cdot center = (temp, CL \cdot y_{temp})$;
 $ASM_{type} \cdot window = \begin{pmatrix} x_1, x_2, SP \cdot P_{left}(\text{from } x_1 \text{ to } x_2), \\ SP \cdot P_{right}(\text{from } x_1 \text{ to } x_2) \end{pmatrix}$);
 $ASM_{type} \cdot active$;
 $V_{up(i)} = \text{result of } ASM_{type}$;
update (local range of CL and SP) and x_2 from ASM_{type} ;
 $type_{old} = type$;
 $i++$;
endwhile

Algorithm 6. – **detecting all vertebrae s from top to down.**
 Input: SP, CL, L_{mid} and $ASM = \{ASM_i/i = 1..6\}$
 Output: $V_{down} = \{V_{down(i)}\}$ – the set of vertebrae, which are located from down to up of spine.
// detect the first vertebrae of spine
 $h = \frac{\sum_{i=4}^6 ASM_i.heigh}{3}$;
 // check value of h with table 1.
 $x_1 = \min(CL.x)$;
 // x_2, x_1 are on CL
 $x_2 = x_1 + h$;
 $temp = \frac{(x_1 + x_2)}{2}$;
 $\mu = getAngle(x_1, x_2, CL)$;
 $type = getTypeOfASM(x_1, x_2, \mu, temp, L_{mid})$;
 //type = 1..6
 $ASM_{type}.center = (temp, CL.y_{temp})$;
 $ASM_{type}.window = \left(\begin{array}{l} x_1, x_2, SP.P_{left}(from\ x_1\ to\ x_2), \\ SP.P_{right}(from\ x_1\ to\ x_2) \end{array} \right)$;
 $ASM_{type}.active$;
 $V_{down(1)} = result\ of\ ASM_{type}$;
 update CL, SP, x_2 from ASM_{type} ;
// detect the remaining vertebrae
 $i = 2$;
 $type_{old} = type$;
 while (data remains on CL) do
 $h = V_{down(i-1)}.height$ // the height of previous located vertebrae was changed after ASM was run.
 // the height of vertebrae is increased by result from ASM models.
 $x_1 = x_2$;
 $x_2 = x_1 + h$;
 $temp = (x_1 + x_2)/2$;
 $\mu = getAngle(x_1, x_2, CL)$;
 $type = getTypeOfASM(x_1, x_2, \mu, temp, L_{mid})$;
 //type = 1..6
 $inter_{disc} =$
 $getInterVertebralDisc \left(\begin{array}{l} V_{down(i-1)}, \\ type, type_{old}, x_1, x_2 \end{array} \right)$;
 //the size of
 //intervertebrae l disc
 if($inter_{disc} > 0$) then
 $x_1 = x_1 + inter_{disc}$;
 $x_2 = x_1 + h$;
 $temp = (x_1 + x_2)/2$;
 endif
 $ASM_{type}.center = (temp, CL.y_{temp})$;
 $ASM_{type}.window = \left(\begin{array}{l} x_1, x_2, SP.P_{left}(from\ x_1\ to\ x_2), \\ SP.P_{right}(from\ x_1\ to\ x_2) \end{array} \right)$;
 $ASM_{type}.active$;
 $V_{down(i)} = result\ of\ ASM_{type}$;
 update (local range of CL and SP)
 and x_2 from ASM_{type} ;
 $type_{old} = type$;
 $i++$;
 endwhile

F. Measuring Angle

The Cobb angle is the angle that is formed by the intersection of two lines, one parallel to the superior endplate of a vertebra above the fracture and another line parallel to the inferior endplate of the vertebrae below the fracture.

We already had the exact position of each vertebra. An algorithm 7 – Measuring Cobb angle algorithm is designed to measure the Cobb angle for each pair of vertebrae. Before we compute the Cobb angle by `getCobbAngle` function, we must know the extreme point of the spine. From the extreme point, we chose a vertebrae (V_i) which is above extreme point and another vertebra (V_k), which is under extreme.

Function `getCobbAngle(V_i, V_k)`
 Input: 02 vertebrae V_i and V_k .
 Output: $angle_{Cobb}$ – the Cobb angle of vertebrae V_i and vertebrae V_k .
 //each vertebra V_i , we have 05 line segments: upper, lower, left, right, center.
 if ($V_i.center < V_k.center$) then
 $V_1 = V_i$;
 $V_2 = V_k$;
 else
 $V_1 = V_k$;
 $V_2 = V_i$;
 endif
 Regressing all points on lower line segment of V_1 to the line L_{lower} ;
 Regressing all points on upper line segment of V_2 to the line L_{upper} ;
 $angle_{Cobb} = angle(L_{lower}, L_{upper})$;
 return $angle_{Cobb}$

Algorithm 7
 Input: the set of all vertebrae $V = \{V_i\} i = \overline{1..M}$. It is ordered.
 Output: the set of Cobb angle
 $point_{center} = \emptyset$;
 for ($i=1$ to M) do
 update $V_i.centerline$ to CL ;
 $point_{center} = point_{center} \cup \{V_i.center\}$;
 endfor
 Regressing all points of CL by least-square linear regress method to polynomial $equ_{CL}(x)$;
 Solve equation $equ'_{CL}(x) = 0$ by numerical method fining a set of extreme points $point_{extreme}$;
 Solve equation $equ''_{CL}(x) = 0$ by numerical method fining a set of inflection points $point_{inflection}$;
 $Cobb = \emptyset$;
 for ($i=0$ to $|point_{inflection}|$) do
 with $point_{inflection}(i)$ get the $point_{extreme}(t)$ corresponding;
 finding the $sup(point_{extreme}(t))$ and $inf(point_{extreme}(t))$ in $point_{center}$;
 $V_1 = V(sup(point_{extreme}(t)))$;
 $V_2 = V(inf(point_{extreme}(t)))$;
 $Cobb = Cobb \cup \{getCobbAngle(V_1, V_2)\}$;
 endfor

The equations of CL are approximated the CL become polynomial by least-square linear regress method, formula (4). We find the extreme points by solving the first-order derivative equation (5).

$$euq_{CL}(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_0 x^0 \quad (4)$$

$$equ'_{CL}(x) = 0 \quad (5)$$

Sometimes, we have some special cases which like Fig. 9; the spine of a patient has the left curve at the top and the right curve at the bottom. This patient needs to be treated two times, one for the left curve and one for the right curve. Then, we must detect and solve the problem, which likes Fig. 9. We must solve the second-order derivative equation to get the number of inflection points, also is the number of the left and right curls of the spine, equation (6). It shows us how many parts of the spine are split.

$$equ''_{CL}(x) = 0 \quad (6)$$

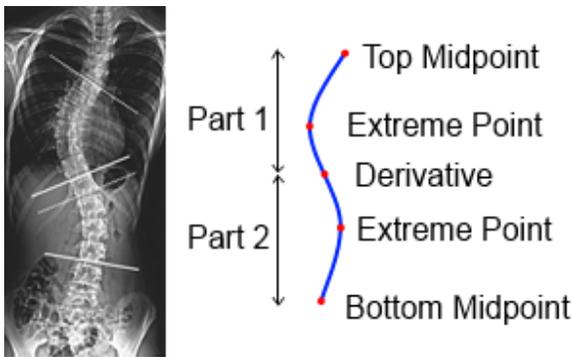


Fig. 9. Illustrate the Curve and Detect Some Important Points to Split the Spine into Parts.

Any pair of vertebra makes the Cobb angle more than 10 degrees is a symptom of scoliosis. We have four scales of evaluation: normal, low warning, high warning, and extreme warning. It bases on the largest Cobb angle of all pairs in this set. If the largest Cobb angle is more than 45 degrees, this is

an extreme warning. If the largest Cobb angle of any pair is between 20 and 45 degrees, this is a high warning. If the largest Cobb angle of any pair is between 10 and 20 degrees, this is a low warning. The last, if all Cobb angles are less than 10 degrees, the spine is normal.

III. RESULTS

We implemented our research on Laptop: chip Intel Core 2 Duo, CPU 2x2.0GHz, with 3GB RAM. The OS system is Windows 7 Professional 32-bit.

The testing database includes X-ray standard images used to illustrate the article, which is posteroanterior radiographs of a female, 16 years old, scoliosis patient.

The rest data in the testing database are from many sources, they are not original images, so their sizes are not uniformed, and many images have not good quality.

Fig. 10 shows two cases of testing. On the left case, the extracted area of the spine does not enough, and there is a noisy area on the left of the spine. The spine in the right case is extracted completely, and there is a little noise in the result.

Table IV shows the result of the first problem. The ratio of spine detection is measured manually because the dataset is not big, we count all pixels of the spine on the original image and all pixels of the extracting area of the spine on result image, then we calculate the ratio by doing the division and round operator. Most images are extracted at least 65% of the spine (excluding cervical vertebrae), and almost the unrecognized area of the spine in the region, which is near the cervical vertebrae. The average accuracy is high. In terms of program execution time, the small image will process quickly, but the bigger the image is, the more running time is increased. The noise magnitude is the area that is extracted but does not belong to the spine; this does not affect much to the vertebrae recognition problem. It's because we apply the ASM model to recognize exactly the bound of vertebrae, then these noises will be automatically ignored.



Fig. 10. Examples of Original Images (Left) and the Results of the First Problem (Right).

TABLE. IV. RESULT OF THE FIRST PROBLEM

Item	Size (pixel)	Recall	Time	Ratio of spine detection	Noise magnitude
1	2208x2688	10	40 mins	100%	very little
2	599x1741	10	20 mins	70%	big
3	1231x2420	10	30 mins	80%	medium
4	768x1024	10	5 mins	65%	big
5	1600x1600	10	10 mins	75%	very little
6	630x630	10	10 s	90%	little
7	250x683	10	7 s	100%	no
8	335x582	10	7 s	70%	medium
9	238x583	10	7 s	90%	medium

TABLE. V. RESULT OF THE SECOND PROBLEM

Item	Size (pixel)	Recall	Time	Auto Evaluation	Manual Evaluation
1	2208x2688	10	5 mins	Extreme	Extreme
2	599x1741	10	2 mins	Extreme	Extreme
3	1231x2420	10	3 mins	Extreme	High
4	768x1024	10	1 min	High	Low
5	1600x1600	10	2 mins	Normal	Normal

The sample models of ASM are trained on the standard image, and then we only test on the first 5 images which have high quality. However, because the size of images is different, when testing on each image, we must resize the models but keep the size's ratio between vertebrae 1 and the image manually. The result of the second problem for the standard X-ray image is good, Table V. Although the detection ratios of the spine are not high in some cases, in the detected region of the spine, vertebrae are extracted correctly, it leads to measure the exact Cobb angle on these vertebrae. The evaluation of the Cobb angle is mentioned in the last part. Our method evaluates on a scale, not on the exact degree of angle, then the lost vertebrae (in the unrecognized region) do not affect much to the evaluation. If these models are applied for low-quality images, the result is not good, and some vertebrae are skimpy or redundant when being extracted. Then, the Cobb angles are much erroneous.

IV. DISCUSSION

The method that we propose not only has good results but also it's simple and effective. In addition, by saving the midpoint, we can determine the direction of the curvature of the spine easily. The advantage of our method, when compared with other method is that our method can determine if a spine which has more than 1 curve, may be considered as scoliosis or not. Moreover, like other methods, our evaluations of the scoliosis are exactly.

A few areas are not handled well, especially the noisy vertebrae. Therefore, the program also needs to be studied more to improve accuracy; for example, the ASM can be combined with other edge detection methods or interpolation methods from the data which are recognized correctly.

The method applied for the second problem should be used for the same standard of an image, such as size, quality to get a good result. Small X-ray image should be applied by models which are trained by small standard image.

We will try to modify this method or find new approaches to improve the performance of this subject.

REFERENCES

- [1] Henry Gray (2000). Anatomy of the human body. Bartleby.com (20th edition).
- [2] McNitt-Gray MF, Huang HK, Sayre JW (1995). Feature selection in the pattern classification problem of digital chest radiograph segmentation. IEEE Transactions on Medical Imaging, 14, 537-547, <https://doi.org/10.1109/42.414619>.
- [3] Luo H, Hao W, Foos DH, Cornelius CW (2006). Automatic image hanging protocol for chest radiographs in PACS. IEEE Trans. Inf. Technol. Biomed, 10, 302-311, <https://doi.org/10.1109/TITB.2005.859872>.
- [4] Ginneken B (2002). Active Shape Model Segmentation with Optimal Features. IEEE Transactions on Medical Imaging, 21, 924-933, <https://doi.org/10.1109/TMI.2002.803121>.
- [5] Cootes TF, Edwards GJ, Taylor CJ (2001). Active Appearance Models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23, 681-685, <https://doi.org/10.1109/34.927467>.

- [6] Cootes TF, Taylor CJ (2004). Anatomical statistical models and their role in feature extraction. *Br. J. Radiol.*, 77, 133–139, <https://doi.org/10.1259/bjr/20343922>.
- [7] Smyth PP, Taylor CJ, Adams JE (1997). Automatic Measurement of Vertebrae 1 Shape using Active Shape Models. *Image Vision Comput.*, 15, 575-581, [https://doi.org/10.1016/S0262-8856\(97\)00006-1](https://doi.org/10.1016/S0262-8856(97)00006-1).
- [8] Smyth PP, Taylor CJ, Adams JE (1999). Vertebrae 1 Shape: Automatic Measurement with Active Shape Models. *Radiology*, 211, 571-578, <https://doi.org/10.1148/radiology.211.2.r99ma40571>.
- [9] Roberts M, Cootes T, Adams J (2003). Linking Sequences of Active Appearance Sub-Models via Constraints: an Application in Automated Vertebrae 1 Morphometry. *The British Machine Vision Conference*, 38, 1-10, <https://doi.org/10.5244/C.17.38>.
- [10] Luc Duong, Cheriet F, Labelle H (2010). Automatic Detection of Scoliotic Curves in Posteroanterior Radiographs. *IEEE Transactions on Biomedical Engineering*, 57, 1143–1150, <https://doi.org/10.1109/TBME.2009.2037214>.
- [11] Long LR, Thomas GR (2001). Landmarking and feature localization in spine x –rays. *J. Electron. Imaging*, 10, 939–956, <https://doi.org/10.1117/1.1406503>.
- [12] T.F. Cootes and C.J. Taylor and D.H. Cooper and J. Graham (1995). Active shape models - their training and application. *Computer Vision and Image Understanding*, 61, 38–59, <https://doi.org/10.1006/cviu.1995.1004>.
- [13] Charles Watson, George Paxinos, Gulgun Kayalioglu (2008). *The Spinal Cord: A Christopher and Dana Reeve Foundation Text and Atlas*. Academic Press (1st Edition).
- [14] Henry Gray (2000). *Anatomy of the human body*. Bartleby.com (20th edition).