Pose Estimation for Vertebral Mobility Analysis using eXclusive-ICA based Boosting (XICABOost) Algorithm

HUANG, Chao-Hui

Abstract—The vertebral pose is critical information in orthopedics. An automated vertebral pose estimation can provide direct supports to medical diagnoses. In this paper, we proposed a vertebral pose estimation based on the given two sets of training patterns. The first set contains the images of vertebrae, in which all vertebral columns are fixed at a proper pose; the second are the images which are cropped with arbitrarily shift and rotation. Based on these two pattern sets, the proposed method can perform template matching. By using exhaustive searching, we will be able to estimate the poses of the vertebral columns on the given x-ray images.

We propose a new approach for extracting critical information from the given training patterns in the problems of classification. In this paper, we use it to estimate the poses of vertebral columns on x-ray images. The proposed method consists of two parts: 1. feature extraction and 2. classification. The first part extracts the unique features from the two given training pattern sets. These unique features are used to support the second part, which is a classifier inspired by the famous AdaBoost.

I. INTRODUCTION

The poses of vertebrae are important information for the diagnosis of many medical conditions, such as back pain and scoliosis. Although there are many relative new technologies, such as Computer Tomography (CT) and Magnetic Resonance Imaging (MRI). Considering the cost of imaging and information acquisition, observing the images obtained from conventional 2-d x-ray imaging is still the most convenient method. However, due to the fact that many of the symptoms of spinal diseases involve complex three-dimensional deformity. Subsequently, there is no trivial way for localizing the vertebra and estimate their poses.

Many computer vision methods were proposed in order to automatically acquire the relevant information of vertebra on digitized x-ray images. The majority methods of state-of-the-art are focusing on segmentation of vertebrae, including shape models [1], [2], [3], [4], point of interests [5], [6], [7], [8] and others.

Kamalakannan et al. suggested an open Directional Gradient Vector Flow (DGVF) snake method for the detection of double-edge of lumbar vertebrae appeared on sagital x-ray images [9]. Their work is based on the fact that an x-ray image actually contains the 3-d structure of the vertebra. Thus, the sharp edges on the top and bottom of the vertebral columns can be observed on the image, especially sagital x-ray images.

Zewail et al. proposed a statistical shape modeling using wavelet-based independent component analysis [10] for detecting the contour of vertebrae. Their method is similar to an active shape model, but focusing on the contour of the vertebrae.

Generalized Hough Transform (GHT) is frequently being used for detecting irregular shape objects. Tezmol et al. introduced an application of GHT for segmentation of cervical vertebrae from x-ray images. One of the major difficulties is that, the contours of vertebrae are versatile and their appearances on x-ray image can be very different from a patient to a patient. This fact can challenge the robustness of any proposed approaches.

Further, compare with thoracic, lumbar and sacral vertebrae, the x-ray images of cervical vertebrae are relative simpler since the cervical region contains less mass of bones. As a result, many researchers have focused on this region [1], [2], [7], [6], [8].

In this paper, we propose a new approach which can find the locations of the vertebral columns and estimate their poses. It considers this topic as a binary classification. That is, the image of a vertebral column is cropped with a proper location and orientation or not. If the answer is yes, then we shall know the location and pose of the vertebral column.

One of the difficulties is, it is necessary to define some features in order to perform the suggested classification. However, these features are not obvious on x-ray images of vertebrae. Especially, on the thoracic, lumbar and sacral regions, the images of many other bones and internal organs are actually overlapped [11]. Thus, it is necessary to develop a method which can extract the features of vertebral images.

The proposed method consists of two parts: one part is eXclusive Independent Component Analysis (XICA). Given two sets of training patterns, XICA can obtain the relevant information of differences and resemblance between the given pattern sets. In this phase, the XICA generates two feature sets of vectors (called exclusive basis sets). These two feature sets contain the critical information that makes one of the given pattern sets different from one to another. The other part is a classification approach inspired by AdaBoost. It uses the information obtained from XICA. Within a series of iteration, a set of weak classifiers are collected and serve as a strong classifier.

In this paper, we first describe the method for preparing the training patterns. Next, we introduce the XICABOost algorithm. Then, in the results session, we show some examples.
II. METHOD

A. Pattern Cropping for Training

We consider the vertebral columns as rigid bodies, jointed by intervertebral fibrocartilage. Thus, in order to obtain the training patterns, we requested medical experts to label the central points of intervertebral fibrocartilage (see Fig. 1). Some examples are shown in Fig. 2.

B. XICABoost Algorithm

1) eXclusive Independent Component Analysis (XICA):

The XICA essentially is a nature extension of Independent Component Analysis (ICA) with the capability of producing exclusive basis sets. While computing the ICA, the algorithm attempts to find independent bases [the independent components. Sometimes it is called mixture matrix in the fields of Blind Source Separation (BSS)] which are capable to represent all of the given training patterns. XICA, however, is slightly different. Given two sets of patterns, the XICA can find the independent bases which are correlated to one of the given training set but uncorrelated to the other. These independent basis sets are so-called exclusive bases. As a result, using the exclusive bases, we are able to identify a specific group of testing patterns by computing the correlations.

For an observed signal consisted of $M$-dimensional variables, $x = (x_1, x_2, \cdots, x_M)^T \in \mathbb{R}^M$, using ICA, it can be represented as

$$x = As,$$  \hspace{1cm} (1)

where $s = (s_1, s_2, \cdots, s_N)^T \in \mathbb{R}^N$ is the vector of the independent latent variables, and $A \in \mathbb{R}^{M \times N}$ is an unknown constant matrix, called the mixing matrix (in sparse representation theories, it is often called a basis set). Note that in many reports of BSS, $N$ is considered as equal to $M$. The goal of ICA is to find an $A$ when only observed signals are given.

In XICA, given two sets of observed $M$-dimensional variables, $x_i \in \mathbb{R}^M$, $i = 1, 2, \cdots, K(x)$ and $y_j \in \mathbb{R}^M$, $j = 1, 2, \cdots, K(y)$, we have

$$x_i = A^{(x)}s_i^{(x)} \quad \text{and} \quad y_j = A^{(y)}s_j^{(y)}, \text{ for all } i, j. \quad (2)$$

where $A^{(x)} = [a_1^{(x)}, a_2^{(x)}, \cdots, a_N^{(x)}] \in \mathbb{R}^{M \times N}$ and $A^{(y)} = [a_1^{(y)}, a_2^{(y)}, \cdots, a_N^{(y)}] \in \mathbb{R}^{M \times N}$ are called exclusive basis sets of the given observed variables $X$ and $Y$, respectively. Denoted by the form of two matrices, (2) can be formatted as

$$X = A^{(x)}S^{(x)} \quad \text{and} \quad Y = A^{(y)}S^{(y)}, \quad (3)$$

where $X = [x_1, x_2, \cdots, x_K^{(x)}] \in \mathbb{R}^{M \times K^{(x)}},$ $Y = [y_1, y_2, \cdots, y_K^{(y)}] \in \mathbb{R}^{M \times K^{(y)}},$ $S^{(x)} = [s_1^{(x)}, s_2^{(x)}, \cdots, s_{K(x)}^{(x)}] \in \mathbb{R}^{N \times K^{(x)}}$ and $S^{(y)} = [s_1^{(y)}, s_2^{(y)}, \cdots, s_{K(y)}^{(y)}] \in \mathbb{R}^{N \times K^{(y)}}.$  \hspace{1cm} (4)
XICA aims to obtain \( A^{(x)} \) and \( A^{(y)} \) based on the nature of the given observed variables.

In order to perform XICA, it is necessary to perform some preprocessing steps: first, we joint the two given training patterns,

\[
\mathbf{\Gamma} = [x_1, x_2, \cdots, x_{K^{(x)}}, y_1, y_2, \cdots, x_{K^{(y)}}] \in \mathbb{R}^{M \times (K^{(x)} + K^{(y)})}. \tag{5}
\]

For the sake of simplifying the notations, the above equation is reorganized as

\[
\mathbf{\Gamma} = [\gamma_1, \gamma_2, \cdots, \gamma_{i}, \cdots, \gamma_{K^{(x)} + K^{(y)}}]. \tag{6}
\]

First, we perform **centering** as the following:

\[
\gamma_i \leftarrow \gamma_i - \langle \gamma_i \rangle_z, \text{ for } i = 1, 2, \cdots, (K^{(x)} + K^{(y)}).
\]

\( \gamma \) is a variable vector representing observed signals distributed according to \( \mathbf{\Gamma} \). \( \langle \cdot \rangle_z \) represent the expected value over \( \mathbf{\Gamma} \).

Next, perform **whitening** as the following:

\[
\mathbf{Z}^{(x)} = \mathbf{VX} = \mathbf{VA}^{(x)}S^{(x)} \text{ and } \mathbf{Z}^{(y)} = \mathbf{VY} = \mathbf{VA}^{(y)}S^{(y)}, \tag{8}
\]

where

\[
\mathbf{V} = \mathbf{D}^{-1/2}\mathbf{E}.
\]

\( \mathbf{E} \) and \( \mathbf{D} \) are the eigenvectors and eigenvalues matrices of \( \mathbf{\Gamma}^T \mathbf{\Gamma} \) such that \( \mathbf{\Gamma}^T \mathbf{\Gamma} = \mathbf{ED} \).

In the case that \( A^{(x)} \) and \( A^{(y)} \) are square matrices, they can be found by using

\[
A^{(x)} = (W^{(x)}V)^{-1} = (V)^{-1}(W^{(x)})^T \text{ and } A^{(y)} = (W^{(y)}V)^{-1} = (V)^{-1}(W^{(y)})^T. \tag{10}
\]

\( W^{(x)} \) and \( W^{(y)} \) are weight matrices corresponding to \( A^{(x)} \) and \( A^{(y)} \). They can be obtained by maximizing the **objective function** over all possible \( (w_i^{(x)}, w_j^{(y)}) \) for all \( i,j \):

\[
\{(w_i^{(x)})^*, (w_j^{(y)})^*\} = \arg \max_{\{w_i^{(x)}, w_j^{(y)}\}} \mathcal{L}(w_i^{(x)}, w_j^{(y)}), \text{ for all } i,j,
\]

where

\[
\mathcal{L}(w_i^{(x)}, w_j^{(y)}) = \left[ (G(w_i^{(x)} \cdot z^{(x)}))_{z^{(x)}} + (G(w_j^{(y)} \cdot z^{(y)}))_{z^{(y)}} \right] \alpha
\]

\[
-\alpha \left[ (G(w_j^{(y)} \cdot z^{(y)}))_{z^{(y)}} + (G(w_i^{(x)} \cdot z^{(x)}))_{z^{(x)}} \right]. \tag{11}
\]

\( \alpha \) is an weight coefficient; \( G(\cdot) \) is used to measure the degree of **non-Gaussianity**. It is usually defined as a **non-quadratic function**. There are few options, for example [12],

\[
G(u) = \begin{cases}
\frac{1}{\alpha} \log \cosh(u/\alpha) - \exp(-u^2/2)
\end{cases}
\tag{13}
\]

To compute (11), one can use the gradient functions:

\[
\Delta w_i^{(x)} = \langle z^{(x)} g((w_i^{(x)} \cdot T z^{(x)})) \rangle_{z^{(x)}} - \langle z^{(y)} g((w_i^{(x)} \cdot T z^{(y)})) \rangle_{z^{(y)}},
\]

\[
\Delta w_i^{(y)} = \langle z^{(y)} g((w_i^{(y)} \cdot T z^{(y)})) \rangle_{z^{(y)}} - \langle z^{(y)} g((w_i^{(y)} \cdot T z^{(x)})) \rangle_{z^{(x)}},
\]

where \( z^{(x)} \) and \( z^{(y)} \) are vector variables representing observed signals distributed according to \( \mathbf{Z}^{(x)} \) and \( \mathbf{Z}^{(y)} \), \( \langle \cdot \rangle_{z} \) represents the expected value over \( \mathbf{Z}^{(x)} \) and \( \mathbf{Z}^{(y)} \), \( g(\cdot) \) is the derivative of the measurement of non-Gaussianity \( G(\cdot) \) as defined in [13].

In order to ensure that the basis sets \( w_i^{(x)} \) and \( w_j^{(y)} \) are orthogonal as possible, we perform decorrelation. First, we obtain a joint matrix

\[
\mathbf{\Omega} = [w_1^{(x)}, w_2^{(x)}, \cdots, w_i^{(x)}, \cdots, w_1^{(y)}, w_2^{(y)}, \cdots, w_j^{(y)}, \cdots, w_1^{(x)}, w_2^{(x)}, \cdots].
\]

(15)

And then,

1) \( \mathbf{\Omega} \leftarrow \mathbf{\Omega}/(\max_i((\sum_j(\mathbf{\Omega})^{2,j}))^{1/2}) \).

2) Repeat \( \mathbf{\Omega} \leftarrow \frac{1}{2}\mathbf{\Omega} \) until convergence. As a result, each of \( w_i^{(x)} \)’s and \( w_j^{(y)} \)’s will be decorrelated to any other and they will be able to provide critical information in the following boosting method for classification.

2) **XICABoost**: The method described in the previous subsection provided two sets of bases, \( w_i^{(x)} \)’s and \( w_j^{(y)} \)’s. They respond to different classes of inputs. That is, given an input \( x \), we can use inner-product to measure its responses on each of classes. Hence, we can define a pool of classifiers,

\[
\mathcal{H} = [h_{1,c_1,d_1}(\cdot), h_{2,c_2,d_2}(\cdot), \cdots, h_{i,c_i,d_i}(\cdot), \cdots, h_{L,c_L,d_L}(\cdot)], \tag{16}
\]

where

\[
h_{i,c_i,d_i}(x) = \log_{10} \left( \frac{w_{i,c_i,d_i} \cdot x}{\|w_{i,c_i,d_i}\|_2} \right). \tag{17}
\]

All \( c_i \)'s and \( d_i \)'s are integer values between \( 1 \) and \( N \) and they are randomly selected in the first place.

The performance of this classifier is based on the chosen \( c_i \)'s and \( d_i \)'s. In this work we use the similar concept of AdaBoost [14] for the selection of these parameters: given a set of training patterns, \( \{(x_i, l_i)\}_{i=1}^K \). Each \( x_i \in \mathbb{R}^M \) represents a training pattern and it is assigned a label \( l_i \in \{-1, 1\} \). \( K \) is the number of total training patterns.

A strong classifier involved a series of chosen weak classifiers. They are selected from \( \mathcal{H} \) over a number of iterations. That is, given an input \( x \), a strong classifier can be defined as

\[
f(x) = \sum_{t} \alpha_t h_t(x), \tag{18}
\]
where $T$ is the number of total iterations, $h_t(\cdot)$ is the chosen weak classifier at the $t$-th iteration, and $\alpha_t$ is a coefficient obtained at the $t$-th iteration.

First, when $t = 1$, each of $K$ training patterns is assigned an initial weight $D_{t,i} = 1/K, i = 1, 2, \cdots, K$.

Next, find the optimal classifier for $t$-th iteration step,

$$h_t(\cdot) = \arg \min_{h_j, c_j, d_j \in \mathcal{H}} \left( \epsilon_t = \sum_{i=1, i \neq h_j, c_j, d_j}^K D_{t,i} \right).$$

(19)

In this step, if the obtained $\epsilon \geq 1/2$ then stop. The classifier pool $\mathcal{H}$ needs to be regenerated. Otherwise, for each $x_i$, compute

$$D_{t+1,i} = \frac{D_{t,i} \exp(-\alpha_t l_i h_t(x_i))}{Z_t},$$

(20)

where

$$\alpha_t = \frac{1}{2} \log_{10} \left( \frac{1 - \epsilon_t}{\epsilon_t} \right),$$

(21)

and $Z_t$ is a normalization factor which keeps

$$\left( \sum_{i=1}^K D_{t+1,i} \right) = 1.$$  

(22)

The above steps are repeated $T$ times. As a result, the strong classifier defined in (18) is obtained.

III. RESULTS

Two results are shown in Fig. 3. Fig. 3(a) shows a relative straight spine image. In Fig. 3(b), the patient was asked to bend the torso thus the medical doctor can observe the angle mobility of the vertebrae.

In order to investigate the performance of the proposed algorithm, we tested it using two sets of pattern images that are cropped using different parameters. The images in the first set are cropped along a horizontal direction without rotation (see Fig. 4(a)). In the second set, the images were rotated arbitrarily at various angles (from $-0.5\pi$ to $0.5\pi$) and cropped. We investigated the output values of the strong classifier described in (18) for these two image sets. The results are shown in Fig. 4 and Fig. 5. The results suggested that the XICABoost is sensitive on image shifting and rotation. As a result, it provides the capabilities for vertebra localization and angle measurement on x-ray images.

IV. CONCLUSIONS

In this paper, we proposed an approach for highlighting the key information of the given training patterns, named eXclusive Independent Component Analysis based Boost (XICABoost). In the application, we showed how to use it to find the locations and estimate the poses of the vertebral columns on x-ray images.

In the experiment, we have run robustness tests. The results suggested that with a proper training patterns, the proposed XICABoost algorithm can be very sensitive on image shifting and rotation. As a result, it can provide useful information for supporting diagnosis of the relevant medical conditions.
(a) 100 image patterns were cropped along a direction.

(b) The output values of the strong classifier described in (18) for the image patterns obtained as described in Fig. 4(a)

Fig. 4. A shifting test for the proposed method. Note that the highest output value of the strong classifier is appeared exactly the center of the cropped vertebral image.

REFERENCES


(a) 100 image patterns were cropped from rotated image (from $-0.5\pi$ to $0.5\pi$).

(b) The output values of the strong classifier described in (18) for the image patterns obtained as described in Fig. 5(a)

Fig. 5. The robustness testing of the proposed method. The image patterns are cropped using various parameters (shift and rotation).


