Segmenting Lumbar Vertebrae in Digital Video Fluoroscopic Images through Edge Enhancement

Shu-Fai WONG and Kwan-Yee Kenneth WONG

Department of Computer Science
The University of Hong Kong
Email: {sfwong, kykwong}@cs.hku.hk

Abstract

Video fluoroscopy provides a cost effective way for the diagnosis of low back pain. Backbones or vertebrae are usually segmented manually from fluoroscopic images of low quality during such a diagnosis. In this paper, we try to reduce human workload by performing automatic vertebrae detection and segmentation. Operators need to provide the rough location of landmarks only. The proposed algorithm will perform edge detection, which is based on pattern recognition of texture, along the snake formed from the landmarks. The snake will then attach to the edge detected. Experimental results show that the proposed system can segment vertebrae from video fluoroscopic image automatically and accurately.

1 Introduction

Low back pain is one of the most common health disorders and its cost can be enormous. There is a general consensus that the diagnosis and the treatment of back pain can be aided by analysing spinal movement. At present, video fluoroscopic imaging provides the only practical method of obtaining images for spinal motion analysis. Unfortunately, the analysis is difficult due to the low quality of the video fluoroscopic images. Figure 1 shows typical video fluoroscopic image of spine.

Figure 1: *A typical video fluoroscopic image of spine.*

In order to study the spinal motion, landmarks of a moving vertebra will be extracted from video fluoroscopic image and will then be analysed. Landmarks are usually the corners of the moving vertebra and are usually extracted manually due to poor quality (noisy and low contrast) of video fluoroscopic images [1]. Operators have to use image enhancement tools to improve the quality first. They will then identify the landmarks by using their expert knowledge. In order to obtain the motion data, operators have to extract at least 20 landmarks from dozens of video fluoroscopic images. The task is time-consuming, tedious and error-prone. In order to reduce the risk of getting inaccurate landmarks, automated procedure is needed.

A wide range of research on automatic extraction of landmarks and segmentation of vertebra have been conducted. In general, there are two main approaches which are widely adopted. The first one is based on template matching and correlation which is simple to implement and easy to understand. In [2], [3], [4], a template comprised the whole vertebra is matched against certain subregion of the image using correlation method. The landmarks are then deduced from the final location of the template. Such approach involves pixel-to-pixel comparison and thus susceptible to changing contrast and pixel intensity of the image. However, such kind of changes is quite common in video fluoroscopic images. In addition, the whole image has to be searched with a large template and thus results in long computational time.

Due to the unreliability and high computational cost of template matching approach, another approach which is based on feature detection is adopted in current research. Features can be corners [5] or shape [6]. These features are detected from the images and the correspondence of them in the images is extracted. Recently, generic shape model and fast feature location method such as generalized hough transform [7] have been proposed to increase the robustness and the speed. However, all methods under this feature detection approach assume the contrast of the image is high such that features can be detected easily. In most circumstances, edges and features have to be manually enhanced and refined before feature location can be done.

In this paper, an image enhancement scheme will be introduced such that there is no need to assume high image contrast and thus the vertebrae can be

extracted with less human intervention. In the proposed system, users have to indicate the rough position of the vertebra. The rough landmarks will form an active contour, or snake, which introduce shape constraints. The image region along the snake is enhanced by the proposed method. The snake will then attach to those region with special features (e.g. corner and edge). The landmarks can then be extracted. The architecture of the proposed system will be described in the next section.

2 System Architecture

The whole system consists of three major components, namely texture feature extractor, texture classifier and boundary detector. The working principle of these components is shown in figure 2.

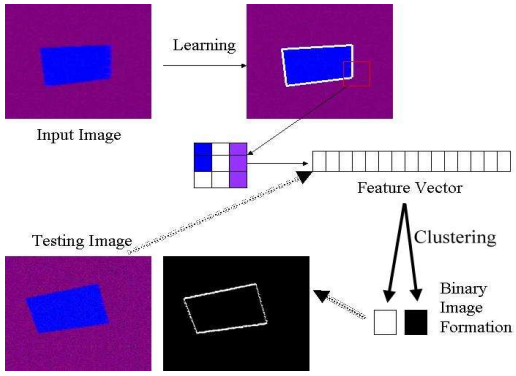

Figure 2: *During the learning phase, both the image and the edge information will be analysed by the system. The system will extract texture feature from the image. By using support vector machine, such feature will be learnt to associate with edge or non-edge class according to the edge information given. During testing phase, the texture feature of testing image will be extracted. The feature will be put into the support vector machine and the associated class will be reported.*

2.1 Texture Description

Spatial properties, like edge and corner, can be inferred by the relationship between the pixels within a small region. For instance, intensity drop along certain direction around a pixel point can be interpreted as the point lie on an edge. The relationship of pixels can be defined by texture.

In the proposed system, we depends on texture analysis to look for an edge. To extract the texture pattern from the image, we adopt Markov Random Field [8], [9], [10] to describe the texture. Under the framework of Markov Random Field, the relationship between pixel and its neighbors is captured as a set of parameters. The relationship or the texture feature can then be used as a heuristic to determine spatial properties of the patch in later stage.

2.2 Texture Clustering

By associating the texture feature to the corresponding texture type, the probability of being certain texture class can also be determined given the feature vector. In the system, there are basically two classes, namely the edge patch and non-edge patch. Support vector machine will be used to associate texture feature and the texture class. During the learning step, the relationship between pixels will be associated with the edge and non-edge classes. For edge class, the feature vector will contain information of slight directional gradient change; while for non-edge class, the vector will show a relatively smooth intensity patch or a non-directional gradient change.

2.3 Texture Segmentation

Once the association is formed by support vector machine, it can be used to classify the incoming feature vector into the edge or non-edge texture classes. The whole testing image will be analysed from pixel to pixel. For each pixel, its texture class will be determined by the classifier after extracting the relationship of the pixel and its neighbors. After performing the classification, a binary image of the texture map, which with value '0' for non-edge and value '1' for edge, will be formed. A snake will be fitted toward the edge texture. The positions of the snake segments will represent the boundary of the target. Landmarks can then be extracted from the snake.

3 Texture Analysis using Markov Random Field

3.1 Markov Random Field

Markov Random Field was developed for texture analysis [11], [12] and [13]. It can be used to describe a texture and make prediction on the intensity value of a certain pixel given the intensity value of its neighborhood. The theories related to Markov Random Field can be found in [8], [9] and [10].

In Markov Random Field, the neighborhood is defined as clique elements. Consider that $S = \{s_1, s_2, .., s_P\}$ is a set of pixels inside the image, and $N = \{N_s | s \in S\}$ is the neighborhoods of the set of pixels. Neighborhood is defined as (1) $s \notin N_s$, (2) $s \in N_r \Leftrightarrow r \in N_s$, and the distance from s to r is bounded. The order of the neighborhood system is determined by the distance between s and r above. A clique $C \subseteq S$ is defined as the subset of pixels such that every combination is a neighborhood system of a certain pixel.

Assuming $X = \{x_s | s \in S\}$ is the random variables (the intensity value) for every pixel inside an image, where $x_s \in L$ and $L = \{0, 1, .., 255\}$. Besides, we have a class set for texture pattern, $\Omega =$

$\{\omega_{S1}, \omega_{S2}, ..., \omega_{SP}\}$ where $\omega_{Si} \in M$ and M is the set of available classes.

In Markov chain analysis, it is stated that we can assume the current state, x_i, is affected by its previous state, x_{i-1} only, instead of infinite number of previous states, $\{x_{i-1}, x_{i-2}, ..., x_0\}$. Similar to Markov Model, there is a local characteristic of the Markov Random Field:

$$P(\omega_s|x_r, r \neq s) = P(\omega_s|x_r, r \in N_s) \quad (1)$$

given that $\forall\omega \in \Omega : P(\omega) > 0$.

The calculation of the density function can be given by Gibbs distribution according to Hammersley-Clifford theorem. Under the theorem, we have the following approximated probability term:

$$\pi(\omega) = \frac{1}{Z} exp(\frac{-U(\omega)}{T}) \quad (2)$$

where Z is a normalising constant:

$$Z = \sum_{\omega} exp(\frac{-U(\omega)}{T}) \quad (3)$$

and T is the temperature constant, which is used in stimulated annealing.

From the equations above, the model of texture is encapsulated by the energy term:

$$U(\omega) = \sum_{c \in C} V_c(\omega) \quad (4)$$

The potential term above, $V_c(\omega)$, describes the configuration potential for a certain class. To have higher probability to be in a certain class, such potential should be as small as possible. In other words, if the relationship between pixels can infer the texture should be in certain class, the above potential should be small after applying the pixels relationship to the equation.

By using Markov Random Field as stated above, the probability of being in a certain texture class can be expressed as the configuration of the cliques, the potential of such configuration depends on the relationship between pixel and its neighborhood.

3.2 Texture Description

In the proposed system, we choose to use first-order neighborhood for simplicity. In other words, the configuration of a clique with have at most 2 elements. By using this neighborhood system and using Bayes decision theorem, equation (4) can be rewritten as:

$$U(\omega, x_i) = V_1(\omega, x_i) + \sum_{i' \in N_i} \beta_{i,i'} \delta(x_i, x_{i'}) \quad (5)$$

the $\delta(x_i, x_{i'})$ is the normalised correlation between pixel at s_i and those at $s_{i'}$.

During the learning phase as described in section 2, the set of $\beta_{i,i'}$ is estimated according to the requirement that the probability of its associated texture class will be maximised. The set of $\beta_{i,i'}$ corresponds to the correlation value and thus represents the configuration of the pixels such that it can be classified as that texture class.

In the system, this set of estimated β will be used as texture feature vector. It will be used as input of support vector machine such that the association between texture feature and texture class can be formed.

4 Texture Clustering using Support Vector Machine

Support vector machine have been widely used in recognition recently due to its non-linear classification power and thus be used to solve complicated recognition problem such as face recognition (e.g. [14], [15]). Given data set: $\{(b_1, y_1), (b_2, y_2), ..., (b_l, y_l)\} \in B \times \{+1, -1\}$, support vector machine can learn to find out the association between b_i and y_i. In the proposed system, the b_i will be the texture feature set $\{\beta_{i,i'}\}$ after texture extraction on the input image and $\{+1, -1\}$ refers to edge and non-edge classes. During learning phase, the support vector machine will be trained to learn the edge and non-edge pattern. During testing phase, the texture feature extracted from the image will be classified by the support vector machine. Mathematical details of support vector machine can be found in [16].

In order to use support vector machine, kernel function should be defined. In the proposed system, gaussian RBF kernel is used:

$$k(b, b') = e^{(\frac{-||b-b'||^2}{2\sigma^2})} \quad (6)$$

According to the support vector theory, the determinant function can be written as:

$$f(b) = sgn(\sum_{i=1}^{l} \alpha_i y_i k(b, b_i) + c) \quad (7)$$

During learning phase, α_i are learnt from data set $\{b_i, y_i\}$ under the following criteria function:

$$max_{\alpha} \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j k(b_i, b_j) \quad (8)$$

In the system, $\{\alpha_i\}$ are learnt through gradient ascent:

$$\alpha_i^{t+1} = \alpha_i^t + \eta \left(1 - y_i \sum_{j-1}^{l} \alpha_j y_j k(b_i, b_j)\right) \quad (9)$$

where η is the learning rate.

During testing phase, equation (7) can be used to determine whether the input pattern is an edge class or not. The output will be an binary image with '1' indicates the edge class and '0' indicates the non-edge class.

5 Texture Segmentation by Snake Fitting

Active contour [17], [18] had been used in pattern location and tracking [19], [20] for a long time. It is good at attaching to object with strong edge and irregular shape. The snake can be interpreted as parametric curve $v(s) = [x(s), y(s)]$.

In the proposed system, the initial position of the active contour is defined by the user. The active contour will move according to the refined energy function:

$$E^*_{snake} = \int_0^1 \{[E_{int}(v(s))] + [E_{texture}(v(s))] + [E_{con}(v(s))]\} ds \quad (10)$$

where E_{int} represents the internal energy of the snake due to bending, $E_{texture}$ represents the texture-based image forces, and E_{con} represents the external constraint forces. The snake is said to be fitted if the E^*_{snake} is minimised.

The above equation is similar to commonly used snake equation but with the energy term $E_{texture}(v(s))$ replaces the original $E_{image}(v(s))$ which means image force. The original term is not used here because it depends on pixel-based edge information (i.e. pixel-to-pixel intensity change). Since the image is of noisy and low contrast, noise will introduce dozens of distracting edges under pixel-based analysis. If the original image energy is used, the corresponding snake will be highly unstable and inaccurate. Instead of using pixel-based edge information, the energy term $E_{texture}(v(s))$ is used in the proposed system. This term represents the energy of texture and is proportional to the negative of similarity of desired texture. This means the energy will be lower near the patch that shows desired texture (e.g. edge texture). Thus, the snake will search for strong edge in the binary texture map, that described in Section 4, along the direction toward the centroid of potential region. It stops at the pixel with strong edge characteristic in the texture map. Thus, the term $E_{texture}(v(s))$ can be interpreted as the texture attractive force and the snake is texture sensitive. Texture represents a patch of pixels instead of a single pixel and texture-based analysis is more tolerant to noise compare with pixel-based analysis. Thus, texture is a much reliable feature than strong edge under pixel-based analysis.

6 Experiment and Result

The proposed system was implemented using Visual C++ under Microsoft Windows. The experiments were done on a P4 2.26 GHz computer with 512M Ram running Microsoft Windows.

6.1 Experiment 1: Synthetic Noisy and Low Contrast Images

In this experiment, the classifier is trained to recognize edge pattern in synthetic images with heavy noise and of low contrast. The total number of samples to be learnt is 1000, half of them is for edge class and half of them is for non-edge class. A texture-sensitive snake is then fitted toward the texture-edge from the initial position close to the real edge. The result is shown in figure 3. It shows that the binary image (texture map) match the edge quite well. The snake can fit toward the edge quite well too. The relative root-mean-square error (i.e. the relative distance between the control points and the real edge) is less than 5% when comparing with ground truth images. The processing time is around 10s where the images with average size 267 x 255 pixels. The experimental result also shows that the system can still fit a snake to the image with error rate lower than 5% under the condition of signal-to-noise ratio equals to 3:1 and the contrast difference is around 10 out of 255 grayscale levels. Under such conditions, common edge detection algorithm will fail in detecting edges.

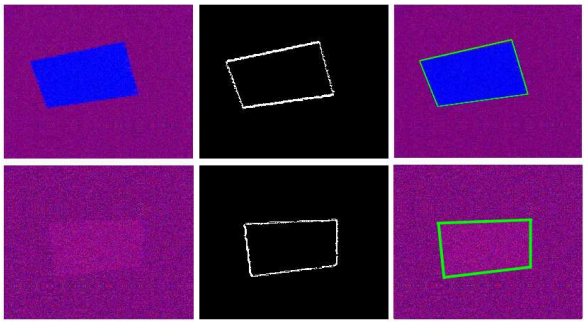

Figure 3: *In this experiment, the target is segmented from the images with heavy noise and of low contrast. The first row shows the result of using noisy image while the second row shows the result of using low contrast image. The images on the left most column show the testing image. The images on the middle column show the binary image after final classification. The images on the right most column show the snake attached to the boundary of the target.*

6.2 Experiment 2: Low Contrast and Noisy Video fluoroscopic Image

In this experiment, the vertebrae has to be segmented from the medical image with poor quality and low contrast. Actually, the image may not be segmented easily manually. In training phase, the total number of samples to be trained is around 1000, half of them is edge and half of them is non-edge. The samples are selected manually and are selected from images with similar illumination and contrast. The learning images and the testing images are randomly

selected from the same video sequence and thus with similar illumination and contrast. The result of segmentation is showed in figure 4. The enlarged image is shown in figure 5. It shows that the snake can fit some of the vertebrae very well. The accuracy cannot be determined here due to no ground truth image provided. If the output is compare with the landmarks marked by a skilled physician, the relative root-mean-square error is less than 3% in average. The processing time is around 18s where the image with size 600 x 450 pixels.

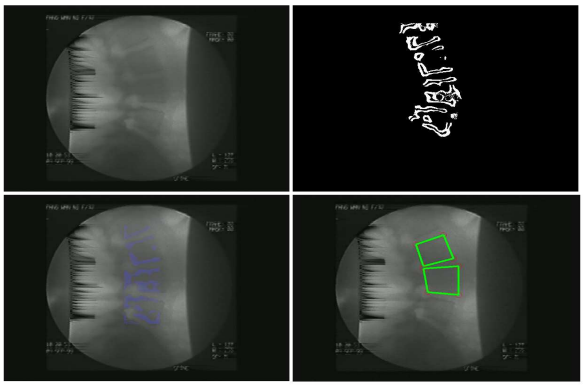

Figure 4: *In this experiment, the vertebrae are segmented from the medical image with low contrast and heavy noise. The vertebrae cannot be observed clearly by human eyes as well. The top left image shows the testing image. The top right image shows the binary image after final classification. The bottom left image shows the fused image constructed from the testing image and the binary image. The bottom right image shows the snake attached to the boundary of the backbones.*
.

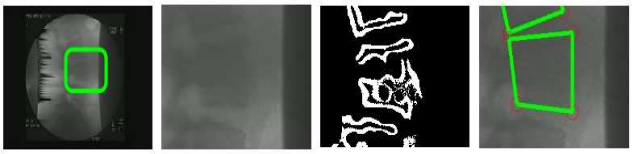

Figure 5: *The corresponding enlarged images of those on the first row and the last column of figure 4.*

6.3 Experiment 3: Comparison with snake approach

In this experiment, commonly used snake approach, which involves using smoothing, edge detection, and snake, is tested. Snake approach and its variation, active shape model, are widely used in analysing video fluoroscopic images (e.g.[6]). In this experiment, the input image is smoothed first so that the noise can be reduced. Sobel edge detection is performed on the resultant image. Snake is finally fitted by replacing $E_{texture}(v(s))$ by $E_{image}(v(s))$, which is the negative of intensity of the edge map, in equation (10). The testing results are shown in figure 6. The results show that if there is no image smoothing have been done, the edge map contains too much strong noise. It seems that the resultant edge map of corresponding smoothed image contains no noise. However, if such resultant image is equalized, it shows that noise does exist actually. In addition, if smoothing is performed on low contrast image before edge detection, the resultant image shows no distinguished edge. This explains why the snake cannot be fitted onto the target appropriately as shown in the last column of the figure.

As shown in the experiment, the performance of snake-based approach mainly depends on the performance of edge detection. Actually, the performance of other approaches, e.g. hough transform approach ([7]), rely mainly on the performance of edge detection too. Without a good edge detection scheme, it seems that it is difficult to extract the landmarks. Common edge detectors, e.g. Sobel and Canny edge detectors, may not be able to detect edge under low contrast and noisy environment. This means snake-based approach and other edge-dependent approaches may not work well automatically. Human intervention is needed in order to refined the edge. In contrast, the proposed system learn the edge pattern in initial stage and can automatically detect edges in testing stage. This greatly reduces human workload in landmark detection by automating the segmentation part.

7 Conclusions

Spine motion analysis on video fluoroscopic images is useful in making diagnosis on low back pain. However, such analysis is difficult due to the low quality of the images. Several approaches, like snake-based approach, have been used to detect the landmarks from medical image and then estimate the dynamic of spine. Such approaches are usually assumes features are clear and can be easily detected. Thus, human intervention is needed in refining such features in order to use such approaches. The proposed system requires less human intervention by automating the edge detection and snake fitting. Operators may need to setup learning pattern and setup initial snake position only. During testing stage, the edge will be detected using pattern recognition automatically and the snake will fit toward the edge accordingly. However, the proposed system is not fast enough for real time analysis as required in applications like real time diagnosis. There is a need to improve the efficiency of the algorithm such that it takes shorter time to extract features and perform classification. In addition, the proposed method is a kind of learning approach that the performance highly depends on the

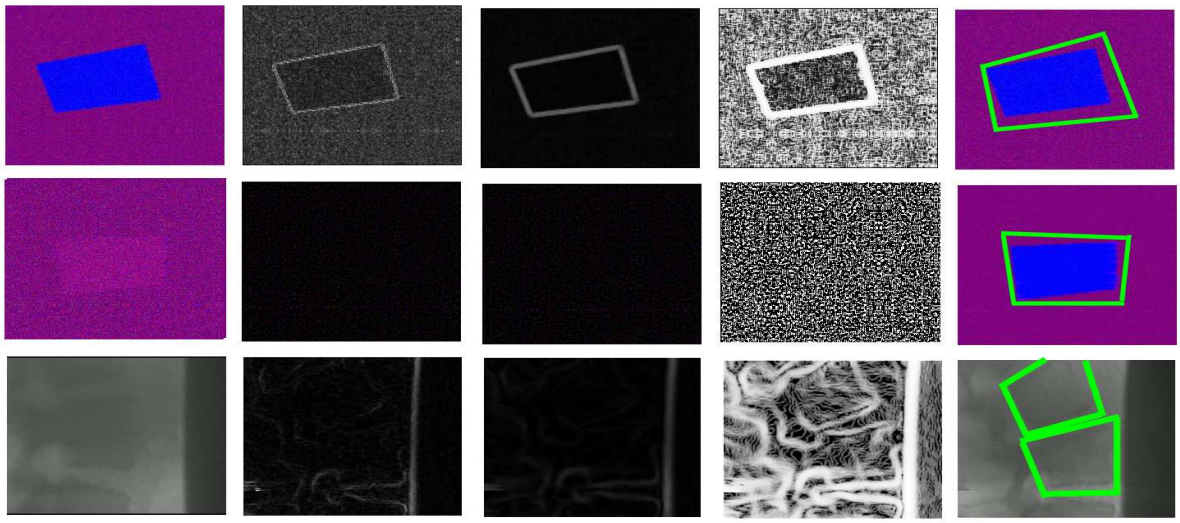

Figure 6: *The results of using snake approach are shown. The first two rows show the result of using artificial noisy and low contrast images. The last row shows the result of using medical image. The images used are the same as those used in previous experiments. The first column shows the input image. The second column shows the edge detection result on unsmoothed input image while the third column shows the edge detection result on smoothed input image. The fourth column shows the equalized result of the resultant image on the third column. The last column shows the snake applied on the the input image with reference to the edge map on the third column.*

learning set. Investigation on applying other learning algorithm and clustering techniques that yield stable performance will be done in the future.

8 Acknowledgement

The work described in this paper was supported by a research grant from The University of Hong Kong.

References

[1] A. C. Breen, R. Allen, and A. Morris, "A digital videofluoroscopic technique for spine kinematics," *Journal of Medical Engineering and Technology*, vol. 13, pp. 109–113, 1989.

[2] C. Simonis, R. Allen, and R. Cloke, "Spatial analysis of the movement of the spine: Application of parallel computing in the field of spine biomechanics," *Transputer Applications*, pp. 35–38, 1993.

[3] J. M. Muggleton and R. Allen, "Automatic location of vertebrae in digitized videofluoroscopic images of the lumbar spine," *Medical Engineering and Physics*, vol. 19, pp. 77–89, 1997.

[4] C. Cardan and R. Allen, "Measurement of spine motion for diagnosis of mechanical problems," *Journal of Simulation Modeling in Medicine*, vol. 1, no. 1, pp. 15–19, 2000.

[5] W. H. Page, W. Monteith, and L. Whitehead, "Dynamic spinal analysis - fact and fiction," *Chiropractic journal of Australia*, vol. 23, no. 3, pp. 82–85, 1993.

[6] P. P. Smyth, C. J. Taylor, and J. E. Adams, "Automatic measurement of vertebral shape using active shape models," *Image and Vision Computing*, vol. 15, pp. 575–581, 1997.

[7] Y. Zheng, M. S. Nixon, and R. Allen, "Automated segmentation of lumbar vertebrae in digital videofluoroscopic images," *IEEE Trans. Medicial Imaging*, vol. 23, no. 1, pp. 45–52, 2004.

[8] J. Woods, "Two dimensional discrete markov random fields," *IT*, vol. 18, pp. 232–240, 1972.

[9] J.Woods, "Markov image modeling," *AC*, vol. 23, pp. 846–850, October 1978.

[10] R. Chellappa and A. Jain, *Markov Random Fields: Theory and Applications*. Academic Press, 1993.

[11] S. Krishnamachari and R. Chellappa, "Multiresolution gauss-markov random-field models for texture segmentation," *IP*, vol. 6, pp. 251–267, February 1997.

[12] R. Dubes and A. Jain, "Random field models in image analysis," *AppStat*, vol. 16, no. 2, pp. 131–164, 1989.

[13] G. Cross and A. Jain, "Markov random field texture models," *PAMI*, vol. 5, pp. 25–39, January 1983.

[14] H. Ai, L. Liang, and G. Xu, "Face detection based on template matching and support vector machines," in *ICIP01*, pp. I: 1006–1009, 2001.

[15] R. Fransens, J. DePrins, and L. J. V. Gool, "Svm-based nonparametric discriminant analysis, an application to face detection," in *ICCV03*, pp. 1289–1296, 2003.

[16] B. Scholkopf, C. J. C. Burges, and A. J. Smola, *Advances in Kernel Methods - Support Vector Learning*. Cambridge, MA: MIT Press, 1999.

[17] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," in *Proc. Int. Conf. on Computer Vision*, pp. 259–268, 1987.

[18] D. J. Williams and M. Shah, "A fast algorithm for active contours," in *Proc. Int. Conf. on Computer Vision*, pp. 592–595, 1990.

[19] A. Blake and M. Isard, *Active Contours*. Springer, 1998.

[20] M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density," in *European Conference on Computer Vision*, pp. 343–356, 1996.