Generalized Dynamic Programming Approaches for Object Detection: Detecting Spine Boundaries and Vertebra Endplates

[a]Guo-Qing Wei, [a]JianZhong Qian and [b]Helmuth Schramm
[a]Imaging Department,Siemens Corporate Research, Inc.
755 College Road East, Princeton, NJ 08650, USA
[b]Medical Solutions, Siemens AG, Erlangen, Germany

Abstract

Object detection employing high-level knowledge is a challenging problem in image analysis. In this paper we propose a dynamic programming approach to address some related issues in this aspect. In particular, we propose to fuse the detection of two curves to form a dual dynamic programming procedure so that spatial relationships between the two curves can be enforced. In another pursuit of applying object level knowledge, we propose to introduce local backward tracing to the forward propagation step in dynamic programming, so that global constraints, containing even unknown parameters, can be imposed in a progressive manner. These approaches are explained in the context of spine landmark detection. Experimental results are presented to show the efficiency of the proposed methods. The methods are extendable to other application domains.

1. Introduction

Dynamic programming (DP) approaches have found applications in many computer vision problems, most notably in stereo matching [2, 5] and curve detection [1, 9, 11, 8]. This paper focuses on the object detection problem. Dynamic programming used in curve detection usually can only handle very primitive knowledge, hardly at the object level. Knowledge at the object level describes object features or properties at a global scale. In this paper, we show how knowledge at the object level can be used to detect objects using generalized dynamic programming approaches.

Boundary detection is usually a part of object detection. The most commonly used prior knowledge in curve detection, which applies to almost all objects, is the smoothness constraint (of boundary), as in the active contour model [10]. The dynamic programming approach for curve detection, either based on the active contour model [1, 9] or the shortest path formulation [11, 8], can handle more efficiently object-specific constraints than the PDE (partial differential equation)-based active contour methods [10, 3]. Anchor points, for example, can be laid, through which the boundary is required to pass. However, for a dynamic programming approach to work, it is essential that the energy function (or cost function) consist of only localized terms, localized in the sense that each term involves only few variables. This restricts the use of prior constraints at the object level. Object level priors usually describe a relationship involving more variables than appear in the localized terms. Therefore, constraints used in previous curve detection methods are usually local in nature. Object-level knowledge was recently used in [6] for object tracking under the assumption of a known point-correspondence between the model and the object. The constraint derived from the model is still at a local level in the above sense. In [12], a symmetry constraint about objects is integrated into the variational formulation for 3D reconstruction. However researchers have noticed that variational approaches are not efficient in enforcing arbitrary constraints [1], for example, constraints involving even unknown parameters, as in this paper. The same is true for the level-set method, which was used to find coupled surfaces in [7].

In the template-based approach for object detection, object-level knowledge was embedded in examples of object appearances, upon which the detection procedure is trained [13, 4]. This is under the assumption that the object shape can be well predicted in the statistical sense. In medical image analysis for diagnosis purposes, it is often the objects (anatomies) that are abnormal that are of interest. Under such circumstances, the shape usually cannot be well modeled by a statistical model. This is the case in the spine landmark detection problem discussed in this paper. Fortunately, even in this case there are still invariant object-level priors that can be used for the detection.

2 Dual Dynamic Programming for Detecting Spine Boundaries

2.1 Problem statement

The object we want to detect in this paper is the spine in digital radiographs. In particular we are interested in de-

0-7695-1272-0/01 $10.00 © 2001 IEEE

tecting spine boundaries and vertebra endplates. These two anatomies, among others, play an important role in quantifying the deformity of pathological spines [14] and thus provide useful information in making diagnostic decisions. Figure 1 shows a spine image and the positions of spine boundaries and vertebra endplates drawn manually, together with the definitions of other terminologies used in this paper.

The detection of spine boundaries and vertebra endplates using either the active contour methods or the active shape methods has to address the following issues. **a)** In many cases, the spine boundaries and endplates are blurred or very weak in contrast due to interferences of other organs; this may cause the active contour methods to get lost or trapped in edges of non-interested organs. **b)** It is not known in advance which part of the spine is to be analyzed, and the number of vertebrae involved is an unknown parameter, too. This, together with the non-predictability of the shape of abnormal spines, makes the active shape model impractical to apply.

In this paper we first detect spine boundaries, and then detect the vertebra endplates with the detected spine boundary as a constraint. The dynamic programming approach is generalized to deal with the aforementioned problems.

2.2 Piecewise linear model

To detect spine boundaries, two priors about spine shape are used to derive the constraints used in the detection. They are: **1)** the orientations of the boundary curves are within a limited angle with respect to the vertical axis (in the human standing pose); **2)** the two boundary curves are nearly parallel to each other.

Since a spine is nearly vertical, we can parameterize the spine boundary by fixed, equally spaced points in the vertical direction. This is like to cut the image plane in vertical directions by a set of equally spaced cut lines. Suppose the y-coordinates of the cutting lines are $\{y_1, y_2, \ldots, y_N\}$, where N is the number of cut lines. The left and right boundaries of the spine can be then represented by the nodal points as $P_L = \{(x_{L,1}, y_1), (x_{L,2}, y_2), \ldots, (x_{L,N}, y_N)\}$ and $P_R = \{(x_{R,1}, y_1), (x_{R,2}, y_2), \ldots, (x_{R,N}, y_N)\}$, respectively. Since y_n's are fixed and predefined, the variables which control the shape of the boundaries are the x-coordinates $(x_{L,1}, x_{L,2}, \ldots, x_{L,N})$ and $(x_{R,1}, x_{R,2}, \ldots, x_{R,N})$. The spacing between the cut lines is chosen such that it reflects the shape scale of interest of the spine.

2.3 Geometric constraints

We use two types of shape constraints on a spine boundary: intra-curve and inter-curve. Intra-curve constraints describe shape constraints on individual boundary curves,

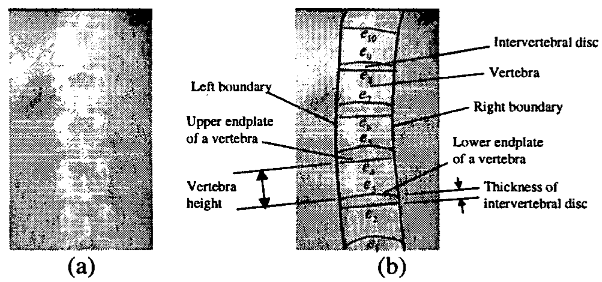

Figure 1: Terminologies. (a) a spine image; (b) annotations.

whereas inter-curve constraints specify the correlations between them.

Under the piecewise linear model, intra-curve constraints place restrictions on **a)** the range of orientations of individual line segments, **b)** the range of inter-segment angles for neighboring line segments, assuring that a smooth spine boundary is generated.

Denote the linear segments in the left and right boundary representation of a spine by $S_L = \{s_{L,1}, s_{L,2}, \ldots, s_{L,N-1}\}$ and $S_R = \{s_{R,1}, s_{R,2}, \ldots, s_{R,N-1}\}$, respectively. The intra-curve constraints, for the left boundary, can be expressed as

$$O_{\min} < O(s_{L,n}) < O_{\max}; \quad n = 1, 2, \ldots N-1 \quad (1)$$

$$\angle(s_{L,n}, s_{L,n+1}) < \gamma_{1,\max}; \quad n = 1, \ldots N-2 \quad (2)$$

$$|\angle(s_{L,n}, s_{L,n+1}) - \angle(s_{L,n-1}, s_{L,n})| < \gamma_{2,\max}, \quad (3)$$

where $O(s_{L,n})$ represents the angle of segment $s_{L,n}$ with respect to the vertical axis, $\angle(.,.)$ denotes the angle between two segments; $(O_{\min}, O_{\max})$, $\gamma_{1,\max}$ and $\gamma_{2,\max}$ are predefined values.

For inter-curve constraints, measurements other than angles will be involved. One inter-curve constraint puts limits on the angles between a segment in the left boundary and the neighboring segments in the right boundary. As has already been noted, this constraint steams from the fact that the left and right boundaries of a spine are almost parallel. The 1^{st} order inter-curve angular constraint can be written as

$$\angle(s_{L,n}, s_{R,n}) < \gamma_{LR,\max} \quad (4)$$

$$\angle(s_{L,n}, s_{R,n-1}) < \gamma_{LR,\max} \quad (5)$$

$$\angle(s_{L,n-1}, s_{R,n}) < \gamma_{LR,\max} \quad (6)$$

where $\gamma_{LR,\max}$ is the maximum angle allowed for local inter-curve segments.

Another inter-curve constraint specifies how vertebra width is allowed to change along the spine axis. From the near-parallelism of the left and right boundaries, a bound is put on the distance changes between two neighboring nodal points on the left and right boundaries. Suppose W_n and W_{n+1} are the distances between corresponding nodal points on the left and right boundaries as projected onto the directions orthogonal to the spine axis. Then the inter-curve distance constraint can be written as

$$d_{\min} < W_{n+1}/W_n < d_{\max} \tag{7}$$

where $d_{\min}$ and $d_{\max}$ are predefined thresholds.

2.4 Hard constraints

Hard constraints can be used to restrict the space of admissible solutions. Anchor points, for example, can be laid. The constraint for a segment s to pas through an anchor point q can be expressed as:

$$d(q,s) < \delta \tag{8}$$

where $d(q,s)$ denotes the distance from point q to segment s, and δ is a tolerance threshold close to zero.

2.5 Optimization

Suppose G(x,y) is the gradient magnitude of a spine image I(x,y). The detected spine boundary is required to maximize the summed gradient magnitude along the path while satisfying all geometric and hard constraints. Formally, the segments $\{ s_{L,n} \}$ and $\{ s_{R,n} \}$ should maximize the following energy function

$$E = \sum_{n=1}^{N-1} \sum_{(x,y)\in S_{L,n}} G(x,y) + \sum_{n=1}^{N-1} \sum_{(x,y)\in S_{R,n}} G(x,y) \tag{9}$$

$$subject\ to: \ \ constraints\,(1) \sim (8)$$

where the first and the second terms correspond to the scores of the left and the right boundaries, respectively. Note that there is no smoothness term in the energy function; we have used inter-segment angular constraint (3) to replace the conventional smoothness constraint, the advantage being that we can explicitly control the shape of the outcome. It has been shown [15] that this optimization can be achieved efficiently by a 2D dynamic programming procedure, which is the integration of two 1D DP procedures; we call this approach the dual-DP method.

3 Endplate Detection Using Global Shape Constraint

Endplates, as a set of unconnected curves, need to be treated collectively, since their positions are constrained by each other. We shall detect all the endplates simultaneously using vertebra shape priors as constraints. A shape prior may also contain unknown parameters that in turn need to be determined.

3.1 Curvature-map projection:

We use a parabolic model to represent an endplate. Therefore, each endplate can be either a curve or a straight line. Since there are three parameters associated with each endplate, we shall first reduce the dimensionality of the problem by using local feature projection. After this projection, each endplate is represented by a single point on the spine axis.

First, a curvature map of the input image is computed. Then for each point p on the spine axis, hypothetic curves are generated to represent the endplate positions and intervertebral discs. Scores of support for the endplates and intervertebral discs are computed as follows. Curvature projections are first made by summing the curvatures along the curves. Then the score supporting the presence of an endplate at p is obtained as the maximum of the projections, and the score supporting the presence of an intervertebral disc is obtained as the minimum of the projections. Since endplates usually appear as ridges, and intervertebral discs as valleys in the curvature map, the maximum curvature projection will have peaks at endplate positions and the minimum curvature projection will have valleys at intervertebral disc positions. For convenience, the minimum curvature projection is negated so that peaks are obtained at intervertebral disc positions. We denote the maximum curvature projection by $C_{\max}(p)$, the minimum curvature projection by $C_{\min}(p)$.

3.2 Dynamic programming to determine endplate positions

According to the characteristics of the curvature projection profiles, an endplate should correspond to a peak on the maximum curvature projection, and between two endplates—one corresponding to the upper endplate of a vertebra, and one corresponding to the lower endplate of the immediately above vertebra—there should be a peak in the minimum curvature projection curve, corresponding to an intervertebral disc. Therefore endplate positions are sought which satisfy these conditions, and at the same time meet all constraints on vertebra shape. The constraints on vertebra shape used in this paper are: **a)** The heights of vertebrae are within a certain range; **b)** The height ratio of neighboring vertebrae is within a certain range; **c)** The ratio of the thickness of the inter-vertebral disc to the vertebra height is within a certain range; **d)** The angle between neighboring endplates should not exceed a certain value. **e)** The vertebra height profile should satisfy a global model.

Ranges of vertebra heights can be computed as being proportional to the vertebra widths, which can in turn be measured from the distance between the left and right spine boundaries previously computed. Suppose the endplate positions to be detected on the spine axis are denoted by $e_1, e_2, ..., e_N,$ where N is the number of endplates. Suppose these endplates are ordered from the lower part to the upper part of the spine body, and e_1 is the lower endplate of the starting vertebra, and e_N is the upper endplate of the ending vertebra, refer to Fig.1. Then the height of each vertebra is

$$h_v = e_{2v} - e_{2v-1}, \quad v = 1, ..., V \tag{11}$$

with the number of vertebrae as V=$N/2$,. The thickness of the intervertebral disc is then

$$t_v = e_{2v+1} - e_{2v} \tag{12}$$

Suppose the orientation of endplate e_v is denoted by $\theta(e_v)$. Then the conditions **a)~d)** can be expressed, respectively, as:

$$h_{\min} < h_v < h_{\max} \tag{13}$$

$$r_{hh,\min} < h_v/h_{v-1} < r_{hh,\max} \tag{14}$$

$$r_{th,\min} < t_v/h_v < r_{th,\max} \tag{15}$$

$$|\theta(e_v) - \theta(e_{v-1})| < \theta_{\max} \tag{16}$$

$$g(h_v) = 0, \qquad \forall v \tag{17}$$

where $h_{\min}, h_{\max}, r_{hh,\min}, r_{hh,\max}$ $r_{th,\min}, r_{th,\max}$, and $\theta_{\max}$ are predefined values. As for the form of $g()$, there exits no physiological standard for describing the relationships among vertebra heights of a spine. However, we found the following form of g() gives a good approximation of this relationship

$$h_v = ka^v + b \tag{18}$$

where the parameters k, a, b are unknown constants for a specific spine, but varies across different spines. Because of the order of the endplates as being from bottom to top, the parameter a will always be less than or equal to 1, meaning that running from the lower to the top part of a spine, the vertebra height should never be increasing. If $a = 1$, equation (18) models vertebrae of constant heights.

Based on the above analysis, the task of endplate detection can be formulated as: to find $e_1, e_2, ..., e_N$ such that a score measuring the degree of support for these positions to be on endplates be maximized while satisfying the constraints (13)~(17). We choose the score as:

$$E = \sum_{i=1}^{N} C_{\max}(e_i) + \sum_{v=1}^{V} \max_{p \in (e_{2v}, e_{2(v+1)})} \{C_{\min}(p)\} \tag{19}$$

where V is the number of vertebrae, which is an unknown parameter. The first term in (19) is the sum of the values of the maximum curvature projections at the endplate positions, while the second term is sum of the values of the minimum curvature projections at the intervertebral disc positions. Since the exact positions of the intervertebral disc are not important to the endplate detection, the second term of (19) is formulated to say that the intervertebral disc should be constrained to lie between an upper endplate and the lower endplate of the immediately above vertebra and to have a peak in the minimum curvature projection.

The maximization of (19) subject to the constraints (13)~(17) is a nontrivial task. What complicates the problem more is the fact that the global constraint (18) involves unknown model parameters.

We propose a generalized dynamic programming approach for the optimization. In the standard dynamic programming approach, constraints like (18) cannot be enforced. Since the model parameters in (18) are unknown, they have to be estimated from the heights of all vertebrae yet to be determined. This means that actually all endplate positions are involved in (18), making the constraint an N-dimensional one. We use local backward tracing to apply the global constraint in a progressive manner as follows.

First, (19) needs to be converted into a form suitable to solve by dynamic programming:

$$\begin{aligned} E \;=\; & C_{s,\max}(e_1) + \sum_{v=1}^{V} [C_{\max}(e_{2v}) + C_{\max}(e_{2v+1}) \\ & + \max_{p \in (e_{2v}, e_{2v+1})} \{C_{\min}(p)\}] + C_{\max}(e_N) \end{aligned} \tag{20}$$

where the score terms are grouped in $V + 2$ groups. The maximization of (20) can be viewed as choosing appropriate variables from $V + 2$ layers, e_1 from the first layer, (e_{2v}, e_{2v+1}) from the following V layers, and e_N from the last layer, such that the total score is maximized. We label the layers by $L_v, v = 0, 1, ..V + 1$ in the sequel.

Instead of directly using endplate positions as the state variables in each layer (except for layers L_0 and L_{V+1}), we employ an equivalent representation (e_{2v}, t_v), where t_v is the thickness of the intervertebral disc between vertebrae v and $v + 1$. The advantage of using (e_{2v}, t_v) to parameterize the state space is that the range of t_v is much smaller than that of e_{2v+1}, reducing significantly the memory requirement in the maximization procedure. From (e_{2v}, t_v), it is easy to get the other endplate position as

$$e_{2v+1} = e_{2v} + t_v. \tag{21}$$

With the new parameterization, the local score for layer L_v in (20) can be written as:

$$E_0(e_1) = C_{\max}(e_1) \tag{22}$$

$$\begin{aligned} E_v(e_{2v}, t_v) \;=\; & C_{\max}(e_{2v}) + C_{\max}(e_{2v} + t_v) \\ & + \max_{p \in (e_{2v}, e_{2v}+t_v)} \{C_{\min}(p)\}; \end{aligned} \tag{23}$$

$$v = 1, ..., V$$

$$E_{V+1}(e_V) = C_{\max}(e_N) \tag{24}$$

The dimension of the dynamic programming would be two if there were no constraint like (18) involved. As was mentioned before, constraint (18) involves all N endplate positions, making the dimension of dynamic programming the same as that of an exhaustive search.

The score maximization of (20) by dynamic programming without the constraints (13)∼(18) can be pursued as follows:

$$S_1(e_2, t_1) = \max_{e_1}[E_0(e_1) + E_1(e_2, t_1)] \tag{25}$$

$$S_v(e_{2v}, t_v) = \max_{e_{2v}, t_v}[S_{v-1}(e_{2(v-1)}, t_{v-1}) + E_v(e_{2v}, t_v)] \tag{26}$$

$$v = 2, ..., V$$

$$S_V(e_N) = \max_{e_{2V}, t_V}[S_V(e_{2V}, t_V) + E_V(e_N)] \tag{27}$$

$$\max_{e_1, e_2, .., e_N} E = \max_{e_N} S_V(e_N) \tag{28}$$

To apply the constraints (13)∼(18), firstly, local constraints (13)∼(16) are used to restrict the parameter ranges over which the maximizations are made. Here it is assumed that the ranges of the first and the last endplates e_1 and e_N are known *a priori*. Denote these ranges by $[e_{1,\min}, e_{1,\max}]$ and $[e_{N,\min}, e_{N,\max}]$, respectively. (Note that the total number of endplates, N, is unknown.) From the height constraint (13), the range of e_2 can be obtained as $[e_{2,\min}, e_{2,\max}] = [e_{1,\min} + h_{\min}, e_{1,\max} + h_{\max}]$. For each $e_2 \in [e_{2,\min}, e_{2,\max}]$, and each $e_1 \in [e_{1,\min}, e_{1,\max}]$, the corresponding vertebra height is $h_1 = e_2 - e_1$. Based on (15), the range $[t_{1,\min}, t_{1,\max}]$ for t_1 can be computed. The propagation from the range $[e_{v,\min}, e_{v,\max}] \times [t_{v,\min}, t_{v,\max}]$ at layer L_v to the layer L_{v+1} is obtained similarly.

For the global constraint (18), local backward tracing is introduced during the forward propagation. For the maximization at layer L_v, back-tracing is performed in the state space for each element $p_{v-1} \in [e_{2(v-1),\min}, e_{2(v-1),\max}] \times [t_{v-1,\min}, t_{v-1,\max}]$ at layer L_{v-1}. This identifies a locally optimal path leading to *each* p_{v-1}. Local optimality is in the sense that elements beyond layer L_v have not yet been considered. From the local path for p_{v-1}, a profile of vertebra heights $H_{p_{v-1}} = \{ h_{p_{v-1}}(i), i = 1, 2, ..., v-1 \}$ can be obtained. Then, for each element $p_v = (e_{2v}, t_v)$ at layer L_v, we check whether the augmented height profile $H_{p_v} = \{ H_{p_{v-1}}, h_{p_v}(v)\}$, with $h_{p_v}(v) = e_{2v} - (e_{2(v-1)} + t_{v-1})$, satisfies the global model (18). To achieve this, an iterative, alternating estimation procedure is adopted to estimate the parameters k, a, b, based on the given profile. Initially, the parameter a is set as $a^{(0)} = 1$, where the super script represents the iteration number. Then, equation (18) becomes linear in k and b, and can be solved for $k^{(1)}, b^{(1)}$. Inserting the obtained $k^{(1)}, b^{(1)}$ into (18), equation (18) can be solved for a to get $a^{(1)}$. With the obtained $a^{(1)}$, new values $k^{(2)}$ and $b^{(2)}$ are solved for. This procedure repeats until convergence is reached. With the obtained fitting parameter, the residual error for each height is checked. A height profile $h(i), i = 1, ..., K$ is said to satisfy the global model if $\forall i$

$$|h(i) - ka^i - b|/h(i) < \delta_h \tag{29}$$

where δ_h is a predefined value. If the height profile satisfies the global model, element p_v is kept. Otherwise, it is discarded. All elements that remain at layer L_v satisfy a global model and are then used in the forward score propagation at the next layer. In this way, the global constraint is ensured in a progressive way, without having to consider all the endplate positions as unknowns at the same time.

Since the range of the last endplate is known, the local and global constraints restrict the number of endplates that can be present between the first and the last endplates.

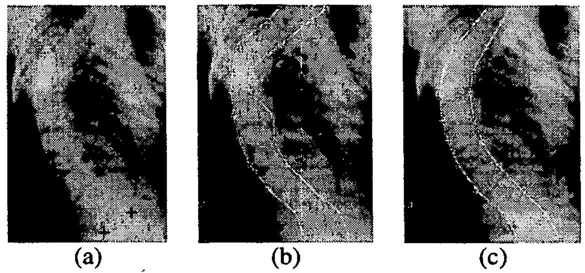

Figure 2: The effects of inter-curve constraints. (a) the original spine image; (b) detected boundary without inter-curve constraints; (c) detected boundary by the dual-DP approach with inter-curve constraints.

4 Experiments

4.1 Initialize the detection

The user needs to select the spine part that is to be analyzed. This is done by clicking on the end points of the first and last endplates of interest. These points also serve as hard constraints for the spine boundary. By allowing several pixels of human errors, the ranges of the first and last endplates are obtained.

4.2 Examples

For the detection of spine boundaries, we first demonstrate how intra- and inter-curve constraints affect the detection results. Figure 2 (a) shows an original spine image of size

(a) (b) (c)

Figure 3: Using global constraints for endplate detection. (a) original image; (b) detected endplates without global constraint; (c) detected endplate with global constraint.

370× 520 pixels, with the four manually selected points marked as crosses. We set an orientation threshold at 60 degrees, and an inter-segment angular threshold at 50 degrees. Figure 2(b) shows the detected boundary curves with the intra-curve constraints only, i.e., no inter-curve ones. It can be seen that although the detected left boundary reflects well the true position of the spine's left boundary, the right boundary has serious errors due to the presence of a spurious strong edge in the middle of the spine. With the inter-curve constraints applied, we get the detected boundary curves shown in Fig. 2(c). By comparing Figs. 2(b) and (c), we can see that the dual-DP method not only finds the correct position for the right boundary, but also improves the localization accuracy for the left boundary.

Next we shall see how the global constraint about vertebra height helps to improve the endplate detection. Figure 3(b) shows the detection results on Fig.3(a), without applying the global constraint. (The endplates are shown as straight lines, with the orientation as the tangent line in the parabolic fitting.) We can see that due to the presence of image disturbances, some endplates are detected at biased positions. When the global constraint is used, the correct endplates are found, see Fig.3(c).

In the above example, it took 12 seconds in a Pentium III 500MHz PC to find the boundary, and 3 seconds to find the endplates.

We tested our algorithm on a small data base consisting of 13 real patient images. It was found that the inter-curve constraints and the global endplate height constraint are especially useful when part of image in the spine area is blurred or noisy. The inter-curve and global constraint help to interpolate automatically the missing information based on the context. Dynamic programming with local constraints only is unable to achieve this.

5 Conclusions

In this paper, we have presented generalized dynamic programming approaches to object detection. We proposed to detect two boundary curves simultaneously by coupling two 1D dynamic programming procedures to form a dual dynamic programming, so that constraints on the spatial relationships between the two curves can be imposed. We proposed to introduce local backward tracing during the forward propagation in dynamic programming as a general technique to enforce global constraints from the object level. Global constraints involving unknown model parameters can be imposed, too.

The methods are applied to the spine boundary detection and endplate detection problems. Promising results are obtained. Future work will be focused on clinical evaluation.

References

[1] A. A. Amini, T.E. Weymouth, and R.C. Jain, "Using dynamic programming for solving variational problems in vision," *IEEE Trans. PAMI*, Vol.12,1990, pp.855-867.

[2] H.H. Baker and T.O. Binford, "Depth from edge and intensity based stereo," *Proc. IJCAI*, , pp.553-572, 1981

[3] L.D. Cohen, "Note: on active contour models and balloons," *CVGIP: Image Understanding*, Vol.53, no.2, pp.211-218, 1991.

[4] T.F. Cootes, C.J. Taylor, D.H. Copper, and J. Graham, "Active shape models—Their training and applications," *Computer Vision and Image Understanding*, Vol.61, pp.38-59, 1995

[5] I. Cox, S. Hingorani, S. Rao, B. Maggs, "A maximum likelihood stereo algorithm," *Computer Vision and Image Understanding*, Vol.63, pp.542-567, 1996

[6] M.-P. Dubisson-Jolly and A. Gupta, "Tracking deformable templates using a shortest path algorithm", *Computer Vision and Image Understanding*, Vol.80,pp.26-45, 2001

[7] X. Zeng, L.H. Staib, R.T., Schultz, J.S. Ducan, "Segmentation and measuremnt of cortex from 3D MR images using coupled surface propagation", *IEEE Trans Imaging Proc.*, Vol.18,pp.927-937, 1999

[8] A.X. Falcao, J.K. Udupa, S. Samarasekea, and S. Sharma, "User-steered image segmentation paradigms: Live wire and live lane," *Graphical Models and Image Processing*, Vol.60, pp.233-260, 1998

[9] D. Geiger, A. Gupta, L.A. Costa, and J. Vlontzos, "Dynamic programming for detecting, tracking, and matching deformable contours," *IEEE Trans. PAMI*, Vol.17, 1995, pp.294-402.

[10] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Proc. ICCV*, pp.321-331, 1988.

[11] E.N. Mortensen and W. A. Barrett, "Ineractive segmentation with intelligent scissors", *Graphical Models and Image Processing*, Vol.60, pp.349-384, 1998

[12] D. Terzopoulos, A. Witkin, M. Kass, "Constraints on deformable models: recovering 3D shape and nonrigid motion", *Artif. Intell.*, Vol.36, 1988, pp.91-123

[13] M.A. Turk and A.P. Pentland, "Face recognition using eigenfaces," *Proc. IEEE Conf. CVPR*, pp.586-591, 1991

[14] B. Verdon, et. al, "Computer Assisted Quantitative Analysis of Deformities of the Human Spine", *Proceedings of Medical Image Computing and Computer Assisted Intervention*, pp.822-831, 1998,

[15] G.Q. Wei, J. Qian, H. Schramm, "A Dual Dynamic Programming Approach to the Detection of Spine Boundaries", *Proceedings MICCAI*, Oct. 2001,