This paper appears in: IEEE International Conference on Image Processing, 1998

A Deformable Model for Human Organ Extraction

Jean Gao, Akio Kosaka and Avi Kak
School of Electrical and Computer Engineering
Purdue University
W. Lafayette, Indiana 47907
jgao@ecn.purdue.edu

Abstract

We present a modification of the well-known snakes algorithm for extracting contours in noisy images. Our modification addresses the issues of selection of the control points on an estimate of the contour and the determination of the weighting coefficients. The weighting coefficients are determined dynamically on the basis of the distance between the control points and the local curvature of the contour. We show results obtained in extracting the liver from cross-sectional images of the abdomen.

1 Introduction

Three-dimensional modeling of liver and the estimation of its volume are powerful tools for diagnosing liver metastases, for the determination of liver regeneration after hepatectomy, and for deciding whether the patient will have enough liver postoperatively to avoid liver failure. Furthermore, liver volume is an important factor when evaluating patients for entire or partial liver transplantation to avoid the possibility of size incompatibility. The 2D segmentation of liver from projection or cross-sectional images is a necessary precursor to 3D modeling.

Previously, Shani *et al.* have proposed knowledge-based algorithms using generalized-cylinder organ models for 3-D structure recognition in abdominal CT scans [1]. Karssemeijer *et al.* [2] described a Markov random field model to segment out the spleen. Bae *et al.* [3] proposed a scheme to detect pulmonary nodules from a set of CT images. Levin [4] explored the detectability of soft-tissue tumors by using multispectral feature space classification.

The state-of-the-art in image segmentation is such that practically all automatic methods available at this time will not work for separating out specific organs from the rest of the image. In current practice, the physicians use either fully-manual or semi-automatic methods. For the fully-manual approach, the user first traces the boundary by marking the high-curvature points and then obtains the edge by interpolation. This method is labor and time consuming, especially when the procedure must be repeated for all the cross-sectional images for the estimation of 3D volume. For the semi-automatic methods, the computer provides the appropriate initial boundary; then using the control points supplied by the user, a more accurate version of the boundary is extracted.

In this paper, we describe an automatic liver boundary extraction method. Our algorithm starts with a newly developed region segmentation algorithm, called the Spedge-and-Medge algorithm, which has been shown to be robust for industrial and robotic applications [5]. This algorithm first eliminates texture appearing in the image by applying a smoothing operator. The algorithm then takes into account the edge information as well as the region homogeneity to generate subregions from the entire image. Next, the algorithm accumulates the segmentated regions that potentially constitute the liver organ based on the geometric properties (location information) and non-geometric properties (intensity information). Once the initial boundary of the liver is generated, a set of representative boundary points is selected on the basis of their curvature properties. These control points are then fed into an energy-minimizing snakes algorithm for a precise delineation of the organ boundary. Note that the control point in our algorithm will, in general, not be equi-spaced. Since the control points are selected on the basis of their curvature properties and with no regard to whether or not they are equi-spaced, we believe we obtain a superior implementation of the snakes approach. This automatic liver contour extraction expedites the next stage of liver volume quantification.

2 Coarse Segmentation

In this section, we will briefly describe the initial segmentation procedure whose boundary output for the liver is later refined by the algorithm presented in

the next section. An example cross-sectional abdominal image that is processed by our system is shown in Fig. 3(a). For the initial segmentation, this image is taken through the following processing steps:

1) The first step uses two thresholds, Low and High, discovered empirically by examining a large number (roughly 100) of cross-sectional images of the type shown in Fig. 3(a) to determine the 99 percentile range for the gray levels of the liver pixels. Subsequently, these two thresholds are applied to every image to discard pixels whose gray levels are obviously outside the liver range. Fig. 3(b) shows the output image obtained when these two thresholds are applied to the cross-sectional image of Fig. 3(a).

2) A 5x5 median filter is then applied to the image obtained from the previous step to reduce the effect of any impulse noise present in the imaging system and to smooth out the textures present in the image. This step makes more robust the performance of the steps to follow.

3) The Canny edge operator is applied to the image obtained from the previous step. This edge detector, like practically all other edge detectors, yields broken and dangling edges. What that means is that the edges by themselves cannot be used for describing closed regions. We first apply an edge linker to the output of the Canny detector to repair small breaks in the edges. The edge linker works on the basis of joining nearby edges provided they are colinear, are separated by no more than two pixels, and if the orientation of the edge end points at the breaks are nearly identical. The partially repaired edge image is fed into the following step.

4) Then, the spedge-and-medge segmentation algorithm is applied to the image to segment out regions inside the body. This algorithm [5] is a modification of the well-known split-and-merge algorithm [9] in which an image is first recursively segmented using a quadtree data structure on the basis of homogeneity of gray levels within the regions represented by the nodes of the quadtree. Next, these regions are merged on the basis of nearness of the gray-levels to the gray levels in other proximal regions and the edge content of each region etc.. The spedge-and-medge modification to split-and-merge consists of taking into account the repaired Canny edges when deciding whether or not to merge two contiguous edges. Fig. 3(c) shows the boundaries of the regions output by the Spedge-and-Medge algorithm.

5) Regions generated by the previous step are analyzed, and geometric and non-geometric attributes are computed for each region. These attributes include: i) Area, ii) Mean gray level, iii) Standard deviation of gray level, and iv) Location (centroid), etc.

6) Using an initial guess for the attribute values, accumulate regions that potentially come from the liver organ. The initial guess is made on the basis of the following two considerations: a) The center of mass of the region should be in the upper left portion of the image since that's where liver is expected to be; and b) the mean gray level in the region should be between the two experimentally-derived thresholds mentioned earlier – Low and High.

7) The liver regions thus obtained are merged together into a single large region. Any holes in this merged region are patched by using one of the standard tools of image analysis.

3 Refinement of Boundary

3.1 The Energy Function

A snake is an energy-minimizing continuous spline whose energy is guided by the image forces that attract it to the wanted image features and the internal energy that serves as a smoothness constraint. The contours slither while always minimizing the energy function, and therefore act like snakes and exhibit dynamic property. This approach to contour extraction from images was first promulgated by Kass *et al.* [6] and since then has found applications in many areas.

The total energy of a spline with parametric description of $S(t) = (x(t), y(t))$ can be described as

$$\begin{aligned} E^*_{snake} &= \int_{snake} E(S(t))\, dt \\ &= \int_{snake} [E_{int}(S(t)) + E_{img}(S(t)) \\ &\quad + E_{con}(S(t))]\, dt \\ &= \int_{snake} [E_{int}(S(t)) + E_{ext}(S(t))]\, dt \qquad (1) \end{aligned}$$

where

$$\begin{aligned} E_{int}(S(t)) &= \frac{1}{2}\left(\alpha(t)\left|\frac{\partial}{\partial t}S(t)\right|^2 + \beta(t)\left|\frac{\partial^2}{\partial t^2}S(t)\right|^2\right), \\ E_{img}(S(t)) &= w_{line}E_{line}(S(t)) + w_{edge}E_{edge}(S(t)) \\ &\quad + w_{term}E_{term}(S(t)), \\ E_{con}(S(t)) &= -K\left(x_1(t) - x_2(t)\right)^2, \\ E_{ext}(S(t)) &= E_{img}(S(t)) + E_{con}(S(t)). \end{aligned}$$

$E(S(t))$ is the local energy associated with the contour at a point defined by the parameter t. This local energy consists of three components: 1) The internal energy, E_{int}, which represents the forces of the spline due to bending; 2) The image energy, E_{img},

which drags the snake to salient features in the vicinity, such as lines, edges, etc.; and 3) The constraint energy, E_{con}, which represents the energy of the spring, with spring constant K, which connects a point x_1 on the contour and the corresponding point x_2 on the spline. The internal energy in turn can be decomposed into two terms, a first-order term and a second-order term. The first-order term, proportional to the first derivative of the contour, makes the snake act like a membrane and the second-order term, proportional to the second-order derivative of the contour, makes it act like a thin-plate. The image energy, E_{img}, can be expressed as a sum of the energies in the different constituents of an image, such as lines, edges, and possibly others. In the expressions above, the line energy at a point defined by the parameter t is represented by E_{line}, the edge energy by E_{edge}, and other possible sources of image energy by E_{term}. The notation here follows the original papers on the subjects [6, 7].

The performance of a snake algorithm is controlled by the coefficients α and β and by the weights w_{line}, w_{edge}, and w_{term}. Unfortunately, there doesn't exist a guideline on how to set values for these coefficients and weights. For obvious reasons, no particular values for these parameters work in all cases. The performance of a snake algorithm, especially its convergence, is also a function of what is used as an initial approximation to the contour, in our case meaning the output of coarse segmentation. We will now present a modified version of the snake algorithm whose convergence is improved by choosing a set of unequally-spaced control points and by using values for α and β that are functions of the local geometry.

3.2 Dynamic Deformable Model

Boundary points are extracted for the potential liver region generated by coarse segmentation (see Step 7 of Section 2). This boundary list is sampled non-uniformly by a recursive algorithm, the placement of the sampling points being determined by the curvature of the boundary. We will not provide the details of this algorithm here for lack of space, but suffice it to say that, starting at a boundary point, the algorithm constructs chords to each of the subsequent boundary points and, as long as the perpendicular distance of the intermediate points to the chord is within a threshold, the chord is made longer. When the constraint on the perpendicular distance is violated, the extension of the chord stops. The boundary point at the end of the chord becomes the next point retained. The boundary points retained in this manner are called the control points.

Let $v_i = (x_i, y_i)$, $i = 0, \cdots, n-1$, be the control points on the boundary obtained in the manner described above. We may refer to these as the state variables for dynamic programming. In terms of these state variables, a discretization of the integrals in Eq. (1) yields

$$E^*_{total} = \sum_{i=0}^{n-1} (E_{int}(v_i) + E_{ext}(v_i)) \tag{2}$$

where

$$E_{int}(v_i) = \frac{1}{2}\left(\alpha_i \left|v_i - v_{i-1}\right|^2 + \beta_i \left|v_{i+1} - 2v_i + v_{i-1}\right|\right)^2, \tag{3}$$

and E_{ext} is given as $E_{ext}(v_i) = -\left|\nabla I(v_i)\right|$, I being the image intensity at stage i.

Dynamic programming is used here to minimize the energy function in Eq. (2) [7]. It is a discrete multistage decision process that varies the control points at each iteration in the direction of a global energy minimum. During each iteration, the optimal energy functional is obtained from all possible shifts of the control points to new positions within an $m \times m$ neighborhood of the current placement of the control points. (In our current implementation, m is 3.) Our implementation decomposes the summation in Eq. (2) into the following form:

$$\begin{aligned} E_{total}(v_1, v_2, ..., v_n) = {} & E_1(v_1, v_2, v_3) + E_2(v_2, v_3, v_4) \\ & + ... + E_{n-2}(v_{n-2}, v_{n-1}, v_n) \end{aligned} \tag{4}$$

where

$$\begin{aligned} & E_{i-1}(v_{i-1}, v_i, v_{i+1}) = \\ & \quad E_{ext}(v_i) + E_{int}(v_{i-1}, v_i, v_{i+1}). \end{aligned}$$

To apply the dynamic processing, a sequence of optimal value functions, $\{S_i\}_{i=1}^{n-1}$, is calculated by

$$\begin{aligned} S_i(v_i, v_{i+1}) = {} & \min_{v_i - 1} S_{i-1}(v_{i-1}, v_i) + E_{int}(v_i) \\ & + E_{ext}(v_i). \end{aligned}$$

The internal energy at the control point v_i is controlled by α_i and β_i. These two coefficients adjust the relative importance to be given to the membrane and thin-plate bending energies. Since the α coefficients control the membrane-like behavior of the contour, if the distance d between two adjacent control points is small, a big value of α_i for that position will cause the boundary to sag into the liver region between those two control points. Therefore, if the distance d between two adjacent control points is small, we'd like the corresponding value of α_i to be also small, and

vice versa. We determined empirically that for best results the relationship between α_i and d_i should be

$$\alpha_i = A\left(1 - e^{-1.67\times10^{-3}d_i}\right) \tag{5}$$

where A is a constant for all stages and is around .003 in the liver image analysis. Fig. 1 shows the relationship.

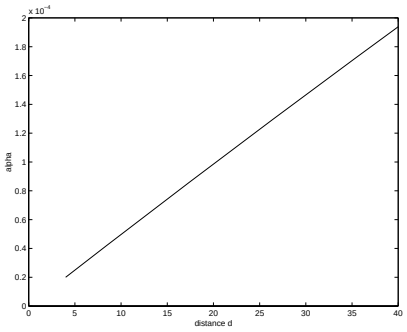

Figure 1: Parameter α_i function.

It is useful for the snake to behave like a thin strip rather than like a membrane. That is, it should try to be a smooth curve or a straight line, but should not contract. The second-order derivative here solves the problem. The coefficient β controls the relative importance of the thin-plate term and the smoothness of the contour. If it is too small, the edge will become coarse. Setting it to zero allows the discontinuity of the second-order and will develop a corner. To those sharp changing contour sections with high curvature, the β should be small to follow the rapid geometrical change; otherwise, those sections will be flattened. The relationship between β_i and the curvature k_i was found to be

$$\beta_i = Be^{-0.01k_i}. \tag{6}$$

The above relationship can also be seen in Fig. 2. In our experiments, we can choose B around 0.002.

4 Concluding Remarks

The robustness of the above method was tested on CT images of abdominal sections containing the liver. Although the complexity of our algorithm is $O(m^3n)$ for each iteration of dynamic programming, the convergence is still fast due to the closeness of the coarse-segmentation contour to the final contour. An important reason for our superior results is that the parameters α_i and β_i are updated continuously in accordance with the changing geometrical properties of the contour locally. Fig. 3 shows the images obtained during the various stages of processing.

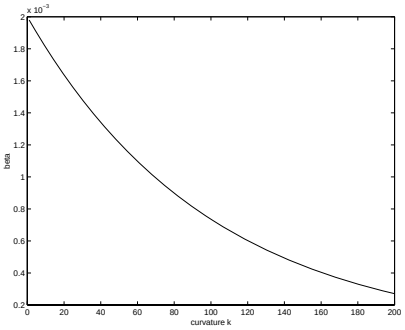

Figure 2: Parameter β_i function.

References

[1] U. Shani, *Understanding 3-D Images: Recognition of Abdominal Anatomy from CAT Scans,*, UMI Research Press, Ann Arbor, MI, 1983.

[2] N. Karssemeijer, L. V. Erning, and E. Eijkman, "Recognition of Organs in CT-image Sequences: A Model Guided Approach," *Comput. Biomed. Res.,* vol.21, pp. 434-438, 1988.

[3] K. T. Bae, M. L. Giger, H. MacMahon, and K. Doi, "Computer-aided Detection of Pulmonary Nodules in CT Images," *Radiology,* vol. 181(P), 144, 1991.

[4] D. N. Levin, A. Hermann, T. Spraggins, P. A. Collins. L.B. Dixon, M. A. Simon, and A. E. Stillman, "Musculoskeletal Tumors: Improved Depiction with Linear Combinations of MR images," *Radiology,* vol. 163, pp. 545-549, 1987.

[5] K. Rahardja and A. Kosaka, "Vision-based bin-picking:Recognition and localization of multiple complex objects using simple visual cues," *1996 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Osaka, Japan, Nov. 1996.

[6] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active Contour Models," *Int J. Comput. Vision*, vol. 1, no. 4. pp. 321-331, 1988.

[7] A. Amini, T. Weynouth, and C. Ramesh, "Using Dynamic Programming for Solving Variational Problems in Vision," *IEEE Trans. Pattern Anal. Machine Intell*, vol. 12, pp. 855-867, 1990.

[8] J. Canny, "A Computational Approach to Edge Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721-741, Nov. 1984.

[9] S.L. Horowitz and T. Pavlidis, "Picture Segmentation by a Tree Traversal Algorithm," *Journal of ACM*, vol. 23, no. 2, pp. 368-388, Apr. 1976.

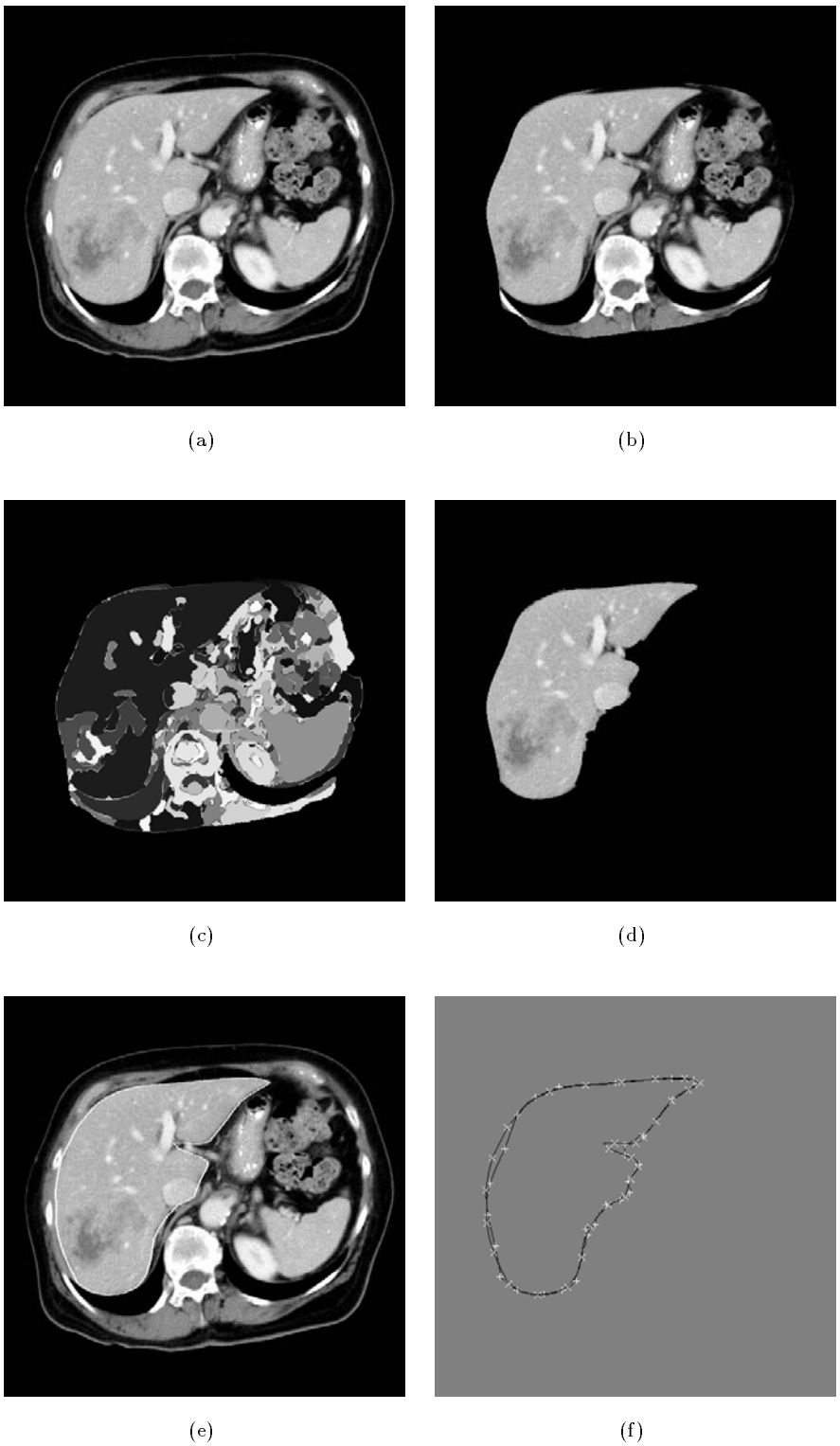

Figure 3: An example of contour extraction. (a) Original image. (b) After gray-level thresholding with Low and High. (c) Result of Spedge-and-Medge segmentation. (d) Region obtained by merging all candidate liver sub-regions. (e) Final segmentation of the liver boundary superimposed on the original image. (f) The final liver contour, represented by x, and the liver contour obtained after just the coarse segmentation, represented by +, are shown here.