

Automatic segmentation in image stacks based on multi-constraint level-set evolution

Ning Dai^a, Hao Liu^a, Yuehong Tang^{b,*}, Jianning Zhao^c and Xiaosheng Cheng^a

^a*College of Mechanical & Electrical Engineering, Nanjing University of Aeronautics & Astronautics, 210016, Nanjing, P.R. China*

^b*College of Science, Nanjing University of Aeronautics & Astronautics, 210016, Nanjing, P.R. China*

^c*Jinling Hospital, Department Orthopedics, Nanjing University School of Medicine, P.R. China*

Abstract. Contour extraction of image stacks is a basic task in medical modeling. The existing level-set methods usually suffer from some problems (e.g. serious errors around sharp features, incorrect split of topology and contour occlusions). This paper proposes a novel method of multi-constraint level-set evolution to avoid above-mentioned problems. Interpolation constraint and deviation constraint are added to evolution process in addition to existing constraints (such as edge points and areas). In order to prevent occlusions, it proposes a method of three-phase level-set evolution. The first phase obtains a rough contour according to selected edge points. The second phase applies an expanding LSE (level-set evolution). Missing edge points in the first phase are added when occlusions probably appear. In the third phase, occlusions are deleted and a refining evolution is implemented. As proved by final experiments, our method can steadily extract contours slice by slice when the shapes of previous contours (contours in the previous slice) are similar to current contours (contours in the current slice). Furthermore, there is no error propagation during the process of contour extraction.

Keywords: Contour extraction, multi-constraint, level-set evolution

1. Introduction

It is a basic technique in CAD to construct free surfaces according to section curves. However, there are still some difficulties in obtaining these section curves. It is still a challenging task to automatically extract section curves in image stacks (such as CT/MRI images) [1].

1.1. Related work

At present, there are some methods of extracting contours in image stacks. These methods can be divided into several main categories as follows: (1) Interactive method [2, 3]. Interactive method is the only method that can extract all the complex contours, but it is time-consuming. (2) Pixel clustering method [4]. This method uses threshold segmentation, region growth and edge detection to obtain regions of interest in tissues. However, problems with region occlusion (or leaking) inevitably appear when this method is used [4, 5]. (3) Contour matching method [6]. It is a method based on geometric

* Address for correspondence: Yuehong Tang, Nanjing University of Aeronautics and Astronautics, P.R. China. Tel.: 025-52075644; Fax: 025-84895781; E-mail: tangyuehong@nuaa.edu.cn.

modeling theories to deal with raw contours which are usually obtained by the pixel clustering method. This method only extracts contours of object from raw contours, so it will lead to the accumulated error. (4) Active contour method. It is a method generally applied in the image processing field. But there are two main problems as to this method, one is construction of initial contour, the other is sensitive degree between active model and initial contour. There are some methods to construct initial contours: (i) Region construction method [5-7]. It is apt to suffer from region occlusions. (ii) Feature extraction method [8-11]. This method usually applies gradient vector field [12] but it does not permit the split of topology. (iii) Contour inheriting method [13, 14]. Previous contours or their deformations are used as initial contours, which makes it difficult to segment joints due to disturbance of other tissues. (iv) Priori knowledge method [15-17]. A priori knowledge of these kinds usually includes pre-defined atlas [15], statistical shape model [16], pre-evaluated shape models [17, 18] and so on. These methods are usually used to aim at some special objects, so they do not get the extensive universality. Besides initialization problem, there are still three problems affecting active contour models: (i) Sharp features of contours. Usually sharp features are abandoned because active contour models serve to smooth curves in the process of curve evolution. (ii) Incorrect split of topology. Though topology split can be regarded as an advantage, incorrect split of topology will generate questionable contours. (iii) Region occlusions (or leaking). It is usually caused by other tissues, noise or weak boundaries.

In the paper, we proposed the modified level-set model according to its flexible ability of change in topology. Different from existing methods that use previous deformed contours as initial contours of active contour models, our method uses previous contours to differentiate edge points as deviation points and interpolation points. These points could be used as constraints of level-set evolution. These constraints are believed to have an effect on following three aspects: (1) improve contour accuracy; (2) prevent incorrect split of topology; (3) overcome noise. As a result, it proves that our method has a higher degree of automation and accuracy.

1.2. Terminology

A current slice is an image in the image stack. Current contours are regions of interest in the current slice. The current contour is usually denoted as C_k . If there are several contours in current slice, they are denoted as $C_{i,k}$ ($i=1, \dots, n$).

A previous slice is an adjoining slice of the current slice. Contours of the interest regions in the previous slice are called previous contours. They are used as heuristic contours when current contours are extracted. The previous contour is usually denoted as C_{k-1} . If there are several previous contours, they are denoted as $C_{i,k-1}$ ($i=1, \dots, n$).

Initial contours are starting contours of active contour evolution. Edge points are those with gray differences between neighboring points in an image. Boundary points are also boundaries of regions.

Deviation points are those from which contours are pushed away in active contour evolutions. Interpolation points are those from which contours are attracted in active contour evolutions. When the evolution is over, contours will pass through interpolation points.

1.3. Overview

After a survey of the related work in this field, the technical process and a description of the flow chart are briefly introduced in Section 2. In Section 3, the edge point selection algorithm is described. Interpolation points are selected according to previous contours. With interpolation points and previous contours, the first level-set evolution is implemented. In section 4, a new level-set evolution

formula with constraints is obtained and an approach is applied to recognize deviation points. In section 5, an inner offset method is introduced to construct initial contours for the second level-set evolution. In Section 6, an occlusion recognition method is introduced to correct the contour obtained in the level-set evolution of the second phase. The corrected contour is used as an initial contour of the level-set evolution of the third phase. Section 7 provides several case-study results of contour extractions for CT images. Some comparisons with other similar algorithms are carried out. The conclusions are summarized in Section 8.

2. Technical process

Our automatic contour extraction process falls into 6 main steps as shown in Figure 1:

(1) Extract edge points and evaluate their normal vectors. In this step, an edge point set P is obtained in a CT image.

(2) Select edge points in outer contours of the interest region according to previous contours. In this step, a set P^d that has been added with missing edge points is obtained.

(3) According to P^d and previous contours, the level-set evolution of the first phase is implemented.

(4) The second-phase level-set evolution is featured by several constraints, which is an expanding process in order to add missing edge points in the level-set evolution of the first phase.

(5) Recognize and correct occlusions in the resulting contours of the level-set evolution of the second phase. Contours corrected are used as the initial contours of the level-set evolution of the third phase.

(6) The level-set evolution of the third phase is used to refine the corrected contours.

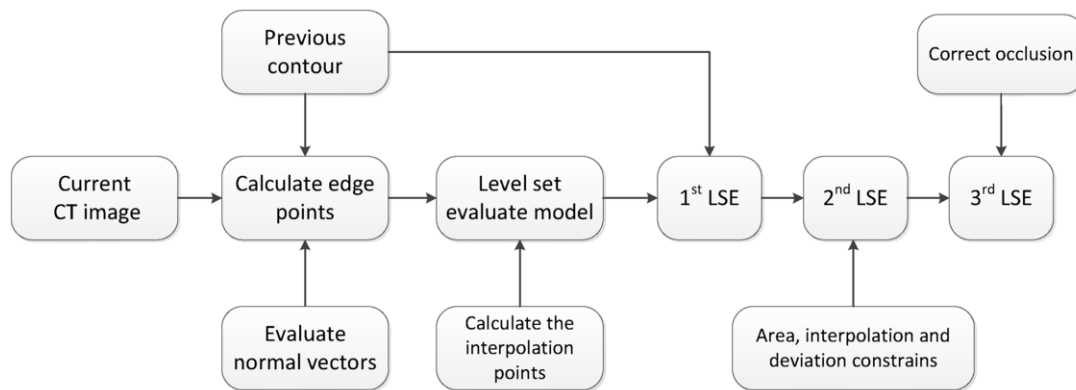


Fig. 1. Contour extraction process.

3. Calculation of the interpolation points

At first, we automatically extract edge points by using an edge detector (e.g. Canny) in the k -th slice and evaluate their normal vectors. Denote all the edge points set by P . If one of these edge points is a boundary point of a region, the normal vector of this point will be toward the outside of the region. We can obtain the result via a steepest gray descending principle [19].

Now we can differentiate interpolation points (outer boundary points of interest region) from the set P . Assuming that the previous contour C_{k-1} has been extracted. Since there is shape similarity between C_{k-1} and C_k , ray projection method could be used to get a corresponding matching point for each point in C_k . A matching point set is denoted as $P^c = \{p_i^c | i=1, \dots, n^c\}$.

Errors often occurred in normal vector evaluation, so some boundary points of interest region may not be included in P^c . This disadvantage is likely result inserious errors of evolving C_k and these errors will probably be passed to the contour extraction process. Consequently, missing edge-points need to be recognized and then added to P^c . Recognition and addition operation can be implemented by region growth process with two conditions in P : (1) normal vectors of grown points and seed points are not in reverse directions; (2) the distance between grown points and seed points is less than a threshold δ . The second condition ensures that a leakage doesn't appear though it still probably skips over some correct edge points. After P^c is added with missing edge points, the new boundary set is expressed as $P^d = \{p^{id} | i=1, \dots, n^d\}$

After obtaining P^d , the first LSE (level set evolution) is implemented by using the previous contour as the initial contours. The first LSE is done in a binary image with points in P^d .

4. Level-set model based on constraints

4.1. Basic level model

Although there are many level-set models [20], we only use the level-set model (DRLSE) [21]:

$$E(\phi) = \mu R_p(\phi) + \lambda L_g(\phi) + \alpha A_g(\phi) \quad (1)$$

where $\mu > 0$, $\lambda > 0$ and $\alpha \in R$ are the coefficients of the energy function. This model can avoid re-initialization. Because of $A_g(\phi)$, the evolution contour can approach boundaries of region of interest by means of the shrinking/expanding mode. Obviously, the above model is an optimization model of multi-objects. For convenience's sake, $R_p(\phi)$ is called regularization constraint, L_g is called edge point constraint and $A_g(\phi)$ is called area constraint.

4.2. Point constraints

Although DRLSE can avoid re-initialization, disadvantages of traditional level-set models remain unchanged, such as noise disturbances, serious errors in sharp regions and the incorrect split of topology. In order to avoid these disadvantages, two constraints are added to the energy model (1): interpolation point constraint and deviation point constraint. As we all know, points in the set P^d are

outer boundary points of regions of interest, so points in the set P^d are used as interpolation points. During the evolution process, contours are attracted to these points.

Noise points are dynamically recognized in the evolution process. We use two ways to recognize noise points. The first method is the normal vector recognition. If an edge point $p \in P$ passes through, we will judge its normal vector. Please note that p has two normal vectors: one is regarded as a boundary point in P , the other is a point in the evolution contour. If these two normal vectors are reverse to each other, p is a deviation noise point. The second one is the reference contour

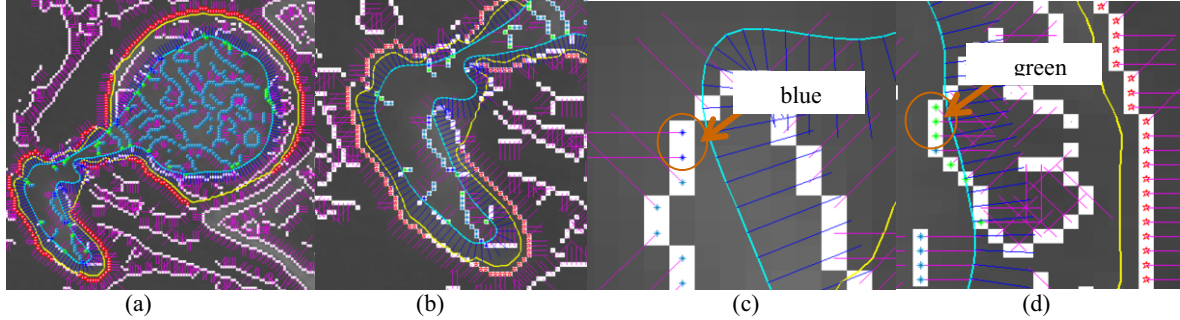


Fig. 2. Illumination of the method of recognizing deviation noise points. (a) the whole image; (b) local amplification; (c) local amplification for deviation points with reverse normal vectors; (d) local amplification for deviation points with similar normal vectors. Yellow curves are reference curves. Cyan curves are evolved curves. White points are edge points. Red pentagons are interpolation points. Deep cyan stars are deviation points in regions enclosed by cyan curves and they are further distance from cyan curves. Blue stars are deviation points around cyan curves and they have reverse normal vectors for cyan curves. Green stars are deviation points around cyan curves and their normal vectors are similar to that of cyan curves. Magenta lines are normal vectors of edge points. Blue lines are normal vectors of cyan curves.

recognition. As to this method, if an edge point p is a deviation noise point, it satisfies the following conditions:

- (1) The distance between p and a reference contour is larger than a threshold σ ;
 - (2) There is no connectivity path in P from p to an interpolation point.
- Figure 2 gives an example to recognize deviation noise points.

4.3. Level-set model of embedding point constraints

In order to control Level-set evolution under point constraints, we add point constraints into constraint terms of an energy-oriented function. Assuming that a point set $P = \{p_1, \dots, p_n\}$. Let $\phi : \Omega \rightarrow \mathbb{R}$ be a LSF (level set function). If we hope that the zero level-set passes through each point in this set, an interpolating function can be defined as:

$$b(\phi) = \begin{cases} |\phi(\mathbf{x})|, & \mathbf{x} \in P \\ 0, & \mathbf{x} \notin P \end{cases}$$

For a point set $Q = \{q_1, \dots, q_n\}$, if we hope that regions enclosed by zero level-set include all points in this set, a leaving function can be defined as:

$$c(\phi) = \begin{cases} \phi(\mathbf{x}), \mathbf{x} \in Q \\ 0, \mathbf{x} \notin Q \end{cases}$$

Denote above two functions by $\chi(\phi)$ and then construct the energy-oriented function:

$$X(\phi) = \int_{\Omega} \delta(\phi) \chi(\phi) dx$$

Where we minimize $X(\phi)$, the zero level-set of ϕ interpolates/deviates corresponding point set. $\delta(\phi)$ is the Dirac delta function. Only when evolution contours approach constraints, interpolation/deviation effects will take place. According to suggestions in [21, 22], $\delta_\epsilon(x)$ is used as an approximation of $\delta(\phi)$. According to variation principle, the time evolution formula to minimize $X(\phi)$ is as follows:

$$\frac{\partial \phi}{\partial t} = -\delta'(\phi) \chi(\phi) - \delta(\phi) \chi'(\phi)$$

When $X(\phi)$ is taken as $B(\phi) / C(\phi)$, $\chi(\phi)$ is taken as $b(\phi) / c(\phi)$. Therefore, we can add interpolation/deviation terms to the energy Eq. (1):

$$E(\phi) = \mu R_p(\phi) + \lambda L_g(\phi) + \alpha A_g(\phi) + \beta B_b(\phi) + \gamma C_c(\phi) \quad (2)$$

So, we can use the following time evolution format to minimize the above energy:

$$\frac{\partial \phi}{\partial t} = -\mu \frac{dR_p(\phi)}{d\phi} - \lambda \frac{dL_g(\phi)}{d\phi} - \alpha \frac{dA_g(\phi)}{d\phi} - \beta \frac{dB_b(\phi)}{d\phi} - \gamma \frac{dC_c(\phi)}{d\phi} \quad (3)$$

5. Construction of initial contours

The evolution of the second phase requires initial contours. The first-phase resulting contour C'_k is an approximation of C_k . Inner offset regions of C'_k are used as initial regions. The inner offset process is implemented by using two key techniques: (1) delete edge points of regions enclosed by C'_k layer by layer; (2) maintain topology. The inner offset process is topology persevering inner offset. In some narrow parts of regions, corresponding inner offset regions become their medial axis. Though

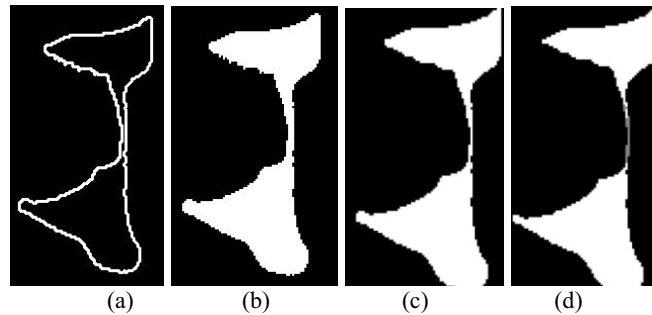


Fig. 3. Example of using inner offset to maintain topology. (a) a contour of hip bone; (b) region enclosed by the contour in (a); (c) the first time to delete edge points; (d) the second time to delete edge points. Gray points are topology maintaining points. If they are deleted, the split of topology will take place. White regions and gray regions form initial regions of level-set evolutions.

such initial contours maintain topology, topology split can be automatically carried out in medial axis positions when existing level-set evolutions are used. One of the aims of our improved level-set evolution is to prevent the incorrect split of topology. Another advantage of our inner offset method is that sharp features become medial axes instead of abandoning them. The closer the initial contours are to current contours, the more easily the accurate current contour will be obtained. Figure 3 gives an example for inner offset to maintain topology.

6. Recognize and correct occlusions

In the energy-oriented Eq. (2), the effect of $A_g(\phi)$ is to expand the zero level-set of LSF ϕ . Since the zero level-set of ϕ has no rigid constraint, occlusions is likely to take place in weak boundary parts. Consequently, we have to check whether there are occlusions when the evolution of the second phase stops. With our method, a contour deformation process is used to recognize occlusions. Let C_k is the second-phase evolution result and C'_k is the first-phase evolution result. Usually, C'_k is a shape approximation of C_k though there are topology differences between each other. Therefore, C'_k can be used as a reference to recognize occlusions in the current contour C_k . For the sake of convenience, temporarily assuming that C'_k and C_k are single contours. That is, C'_k is a close curve, and so is C_k .

6.1. Occlusion recognition based on ray projection

For recognizing occlusion, C'_k is evolved to C_k by a ray projection process, as shown in Figure 4. When the projection evolution stops, the evolved curve is probably an approximation of C_k . Since C'_k is a good approximation of C_k in non-occlusion parts, the result of projection evolution can perfectly approach C_k in non-occlusion parts while errors take place in occlusion parts. Since errors in occlusions don't affect our occlusion recognition and corrections, we can use the result of projection evolution to replace C_k . That is, we actually recognize occlusions in the projection result of evolution.

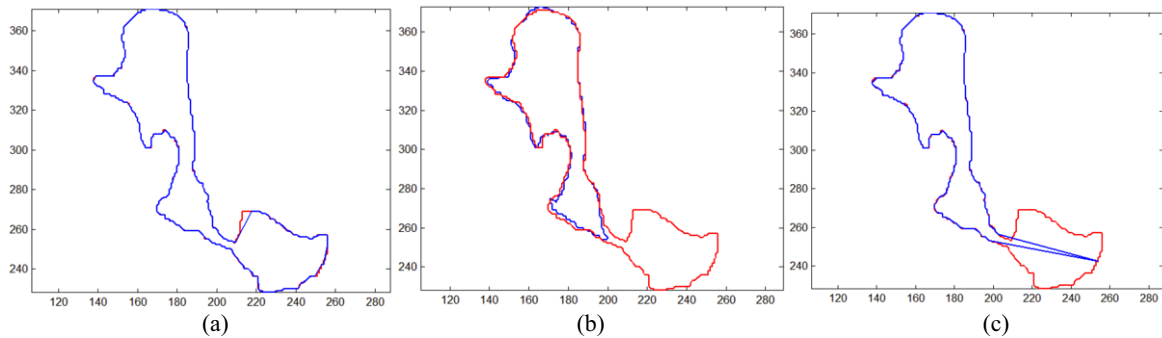


Fig. 4. Process of contour evolution by using ray projections. (a) Evolving contour (blue) and current contour (red); (b) the first evolution of the reference contour; (c) the second evolution of the reference contour. When the evolution process finishes, the blue contour is used to replace the red curve for recognizing occlusion.

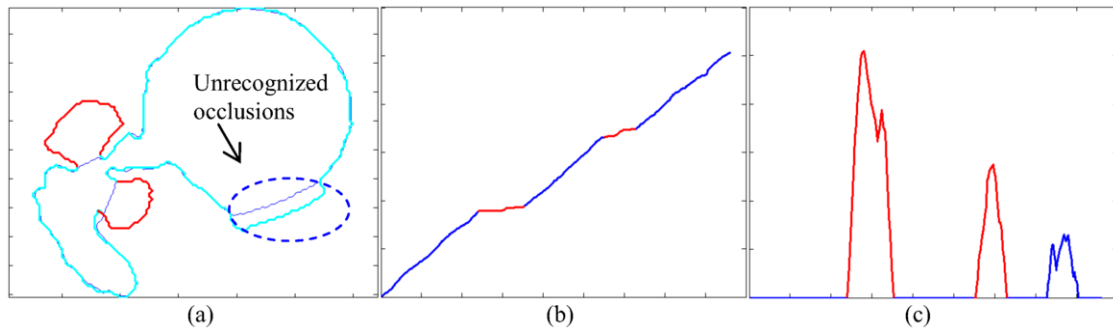


Fig. 5. Curve evolution and parameterization. (a) Contours and occlusions. Blue curve is an evolving contour (C'_k). Cyan curves are current contours (C_k). Red curves are occlusions. (b) index-uparc curve. Red curves are parts corresponding to occlusions; (c) index-distance curve. Red curves are parts corresponding to occlusions.

In the projection evolution, two parameterization curves (the index-uparc curve and the index-distance curve) are constructed (shown in Figure 5). These two curves respectively represent the arc length parameter on the previous contour and the movement distance moving from initial position to present position for every point in the evolved curve. In Figure 5(b), there are some curve segments that are close to horizontal line segments. These curve segments correspond to occlusions in the evolved curve. However, there are probably many “horizontal” fragments in C_k . Several fragments probably correspond to an occlusion in C_k . By using a contour track process in the index-distance curve, these fragments corresponding to the same occlusion can be united. So, the occlusion is recognized. Figure 5(a) shows recognized occlusions. In Figure 5(a), we can find that some occlusions are not recognized. Area constraint and our deviation constraints will enlarge recognition region and enable them to be recognized by our method of occlusion recognition. Such an example is used only because we want to demonstrate the robustness of the method. From the point of view of the similarity of shape, such a part can't be regarded as an occlusion indeed because there is little parallel distance between the occlusion curve and its corresponding curve segment in the reference curve.

6.2. Connect contour

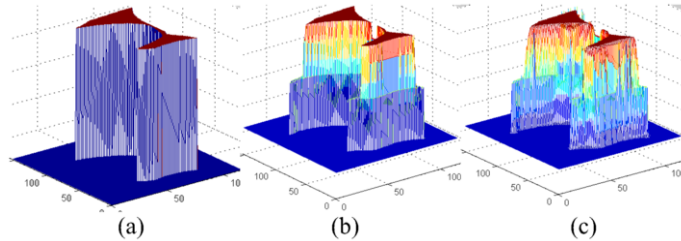
Medial axes of each other are used to connect C'_k and C_k on the assumption that they have the same topology structure. Let's assume that regions enclosed by C'_k and C_k are respectively Σ'_k and Σ_k . M'_k and M_k are medial axis curves of the two regions. Let $\Sigma'_{m,k} = \Sigma'_k \cup M_k$ and $\Sigma_{m,k} = \Sigma_k \cup M'_k$. There are probably non-overlapping regions between Σ'_k and Σ_k , which results in end-points of M_k are out of Σ'_k and end-points of M'_k are out of Σ_k . So, we iteratively delete these end-points M_k that are not in Σ'_k for $\Sigma'_{m,k}$. A similar process is done for $\Sigma_{m,k}$. With this method, two regions with the same topology structure are obtained. After extracting contours of the two regions, we can obtain $C'_{m,k}$ and $C_{m,k}$. By using the occlusion recognition process in the above sub-section, we can recognize and correct occlusions in C_k .

C_k without occlusion can be regarded as a good approximation of the accurate contours. Consequently, it can be used as an initial contour of the level evolution of the third phase. In the evolution, we ignore area constraints and deviation constraints by setting their coefficients as zero in the evolution format (3).

7. Results and analysis

7.1. LSF surface evolutions

Now, let's illustrate how our LSF constraints prevent topology split and preserve sharp features by means of LSF evolution graphics. In our implements, added constraints are likely to result in the breakage of regularity of original LSE, which leads to the occurrence of some odd topologies and wrinkles (see Figure 6(f)). In Figures 6 (d) and 6(e), some spiculate peaks can be found. These spiculate peaks correspond to odd little topologies and wrinkles in zero level-set of LSF. However, spiculate peaks can be eliminated by evolving several times by original LSE after interpolation evolution is over. By comparing Figure 6 with Figure 7, it is easy to see that original LSE has better regularity. However, better regularity is at the expense of abandoning sharp features together with the incorrect split of topology.



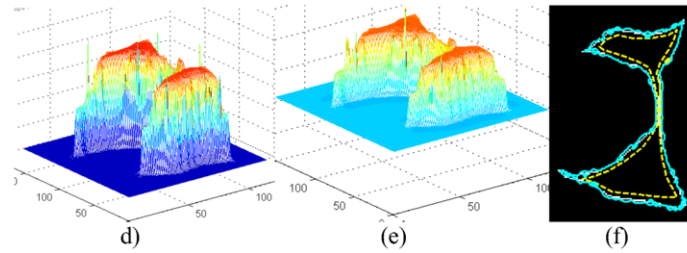


Fig. 6. LSF evolution surfaces and zero level-set of DRLSE with interpolation constraints. (a) Initial surface; (b) 2nd step; (c) 5th step; (d) 8th step; (e) 15th step; (f) zero level-set of LSF in 15th step.

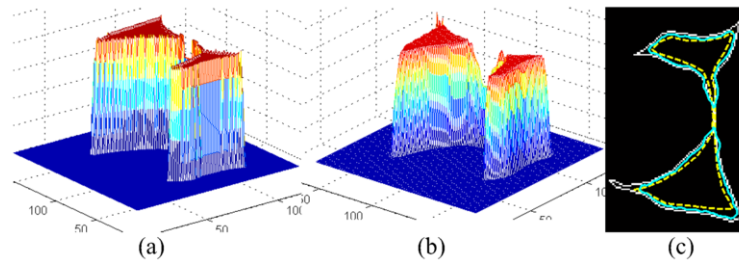


Fig. 7. LSF evolution surfaces and zero level-set of DRLSE. (a) 3rd step; (b) 15th step; (c) zero level-set of LSF in 15th step.

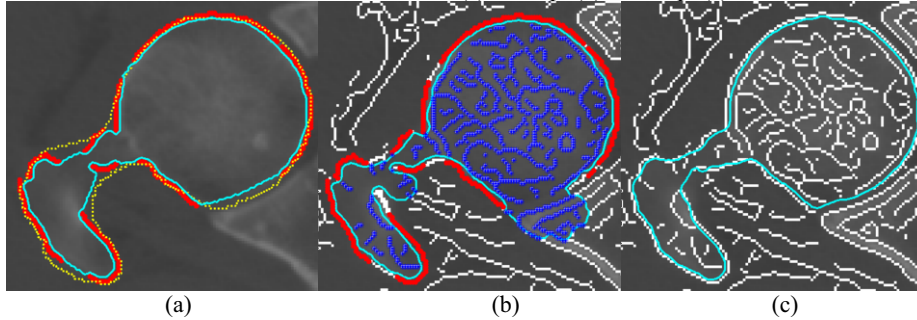


Fig. 8. Segment a slice of a femur. (a) Extracted boundary points (red), the first-phase evolution contour (cyan) and previous contour (yellow). (b) Interpolation points (red), deviation points (blue) and the second-phase evolution contour (cyan). White points are strengthened edge points. (c) Resulting contour (cyan).

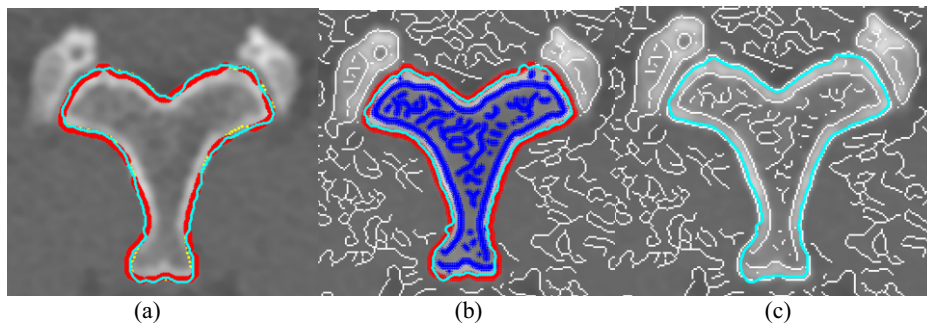


Fig. 9. Segment a part of spine. (a) Extracted boundary points (red), the first-phase evolution contour (cyan) and previous contour (yellow). (b) Interpolation points (red), deviation points (blue) and the second-phase evolution contour (cyan). White points are strengthened edge points. (c) Resulting contour (cyan).

7.2. Results for current slices

Let's give two examples that illustrate the feasibility of our automation extraction method in Figures 8 and 9. In Figure 8(a), it can be found that there are some serious errors between previous contours and current contours (refer to yellow curves). After the first-phase evolution, we can obtain the cyan curve which is a good approximation of the current contour. However, some parts of current contours are still missing due to missing interpolation points. Figure 8(b) shows the result of the second-phase evolution. Missing parts are added while occlusions take place. In Figure 8(c), we delete occlusions and refine the curve after deleting occlusion by using the third-phase evolution. In Figure 9, we segment a part of a spine. Segmentation of spines is usually considered very difficult due to gray distribution unbalance. Since we use edge detectors to strengthen edge points, our algorithm can extract an accurate contour in the first evolution. However, the first-phase evolution can't judge whether or not such a contour is accurate. Consequently, the following two phases of evolution are still required. The last resulting contours in Figures 9(b) and 9(c) prove the accuracy of our algorithm.

7.3. Comparisons

We compare automation degree and accuracy between our algorithm and Djemal, et al.'s algorithm [14]. The essence of Djemal, et al.'s algorithm is to predict C_k according to C_{k-2} and C_{k-1} , followed by the application of the dilation of C_k as the initial contour, as shown in Figure 10. Two

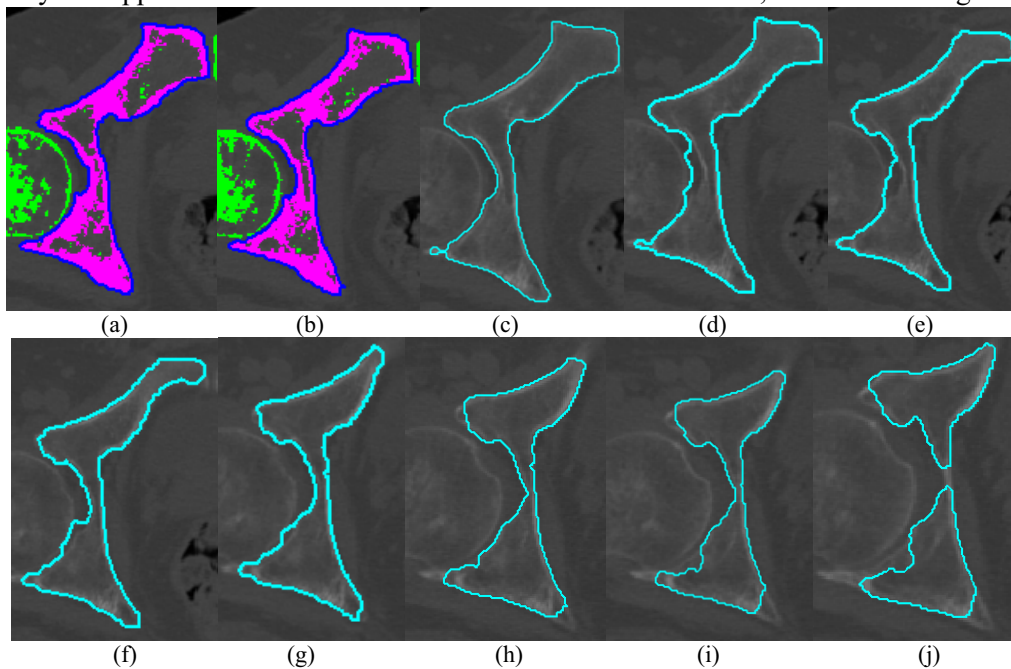


Fig. 10. Extract contours in CT slices by using Djemal, et al.'s algorithm. (a) 55th slice; (b) 54th slice; (c)-(j): The 53rd slice-the 47th slice. Contours in the 55th slice and the 54th slice are extracted in advance. In these two slices, contours and interest regions are emphasized by blue and magenta. They are used as seed contours. Contours in the 53rd slice-the 47th slice are automatically extracted.

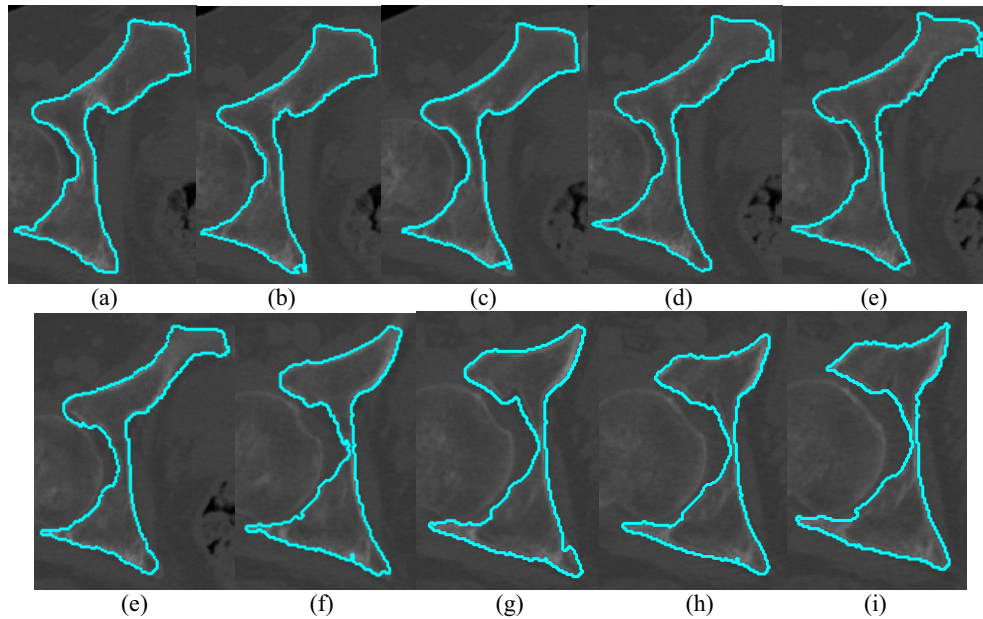


Fig. 11. Extract contours in CT slices by using our algorithm. (a)-(i): contours of a hip bone from the 53rd slice to the 44th slice.

contours are extracted firstly from the 55th slice and the 54th slice respectively. The two contours are accurate contours. Contours from the 53rd slice to 47th slice are extracted by using Djemal, et al.'s algorithm in Figures 10(c) to 10(j). We use the manual extraction contour in 54th (see Figure 10(b)) as seed contour. All contours from the 53rd slice to 44th slice are automatically extracted by using our algorithm in Figure 11.

When Djemal, et al.'s algorithm automatically extract contours from 53rd slice to the 47th slice, errors of extracted contours are accumulated in Figures 10(c)-10(j). At last, the incorrect split of topology takes place and the automatic contour extraction process can't continue. From Figure 11, we can find that our algorithm can automatically extract contours from 53rd slice to 44th slice, sharp features haven't been ignored and there is no topology split in any slice. It is proved that there is no error propagation in our algorithm during contour extraction process

The idea of our algorithm is also similar to that of Liu, et al.'s algorithm [6]. However, Liu, et al.'s algorithm can only deal with raw contours while our algorithm can deal with scattered edge points. Furthermore, their paper also suggests a hybrid seed contour strategy, which will lead to a low degree of automation.

7.4. Application in CT Images

Figure 12 gives an example to segment a hip bone from an image stack with 110 slices. Slices from the 1st to the 101st include the hip bone. Firstly, we selected the seed slice in which we manually extract contour before using our method. The 55th slice is used as the first seed slice. So, slices from the 56th to the 101st can be automatically segmented. After the backward segmentation is completed, a frontward segmentation (from 54th to 1st) is done. There is a pair of adjacent slices in which contour shapes are very different. Such differences have an effect on our method's result. When we only reset the 29th slice as a new seed slice, our algorithm can continue to work from 54th to 1st slice. We can automatically get high quality tissue contours by using our algorithm see Figures 12(a) and 12(b).

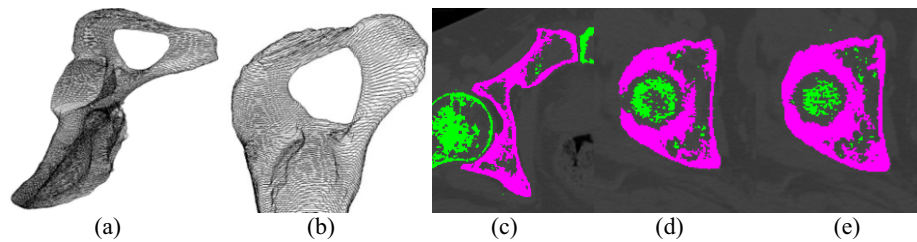


Fig. 12. Example of extracting hip bone contours. (a) Contours in all slices; (b) Local magnification; (c) the first seed slice; (d) the 30th slice and (e) the 29th slice set as a seed slice because of large contour shape difference between the 30th slice and the 29th slice.

8. Conclusions

This paper proposes a method to automatically extract contours in image stacks slice by slice. The method is composed of three-phase level-set evolutions. Compared with single level-set evolution, our method has two advantages: (1) It can recognize weak boundaries while a single level-set evolution usually cannot work well; (2) It can overcome occlusion problem which we suffer from a single level-set evolution.

Automatically extracting the contours from region of interest is a challenging work. In this paper, we spend more time extracting the contours by using three times level-set evolutions in order to improve the contours quality. Our future work will further optimize the calculation speed.

Acknowledgment

This work was supported by the NUAU Fundamental Research Funds under Grant NZ2013201.

References

- [1] Feng Liu, Wei Liu, Xiaoyi Li, Shiyu Wang and Jiajia Fan, Current status of biomedical imaging technology and method in China: A review, *Biomedical Engineering Research* **3** (2014), 64–73.
- [2] M. Färber, J. Ehrhardt and H. Handels, Automatic Atlas-based contour extraction of anatomical structure in medical images, in: *Computer Assisted Radiology and Surgery Proceedings of the 19th International Congress and Exhibition*, Heinz U. Lemke, Kiyonari Inamura and Kunio Doi, eds., Elsevier, USA, 2005, pp. 272–277.
- [3] T.M. Sketch Snakes, Sketch-line initialized snakes for efficient interactive medical image segmentation, *Computerized Medical Imaging and Graphics* **32** (2008), 331–352.
- [4] M.G. Jinich, P. Meer and V. Medina, Robust retrieval of 3D structure from image stacks, *Medical Image Analysis* **3** (1999), 21–35.
- [5] F. Shi, J. Yang and Y. Zhu, Automatic segmentation of bladder in CT images, *Journal of Zhejiang University, Science A* **10** (2009), 239–246.
- [6] S. Liu and W. Ma, Seed-growing segmentation of 3-D surfaces from CT-contour data, *Computer Aided Design* **31** (1999), 517–536.
- [7] A. Daoudi, S. Mahmoudi and M.A. Chikh, Automatic segmentation of the left atrium on CT images, in: *STACOM 2013*, O. Camara, et al., eds., Springer-Verlag, Germany, 2014, pp. 14–23.
- [8] B. Li and S.T. Acton, Automatic active model initialization via poison inverse gradient, *IEEE Transactions on Image Processing* **17** (2008), 1406–1420.

- [9] M. Charfi, Using the GGVF for automatic initialization and splitting snake model, in: 5th International Symposium on I/V Communications and Mobile Network (ISVC), Mohamed Essaaidi, et al., eds., IEEE Press, U.S.A., 2010, pp. 1–4.
- [10] C. Tauber, H. Batatia and A. Ayache, Quasi-automatic initialization for parametric active contours, *Pattern Recognition Letters* **31** (2010), 83–90.
- [11] H. Liu, J.N Zhao, N. Dai, H.B. Qian and Y.H. Tang, Improve accuracy for automatic acetabulum segmentation in CT images, *Bio-Medical Materials and Engineering* **24** (2014), 3159–3177.
- [12] C.Y. Xu and J.L. Prince, Snakes, shapes, and gradient vector flow, *IEEE Transactions on Image Processing* **7** (1998), 359–369.
- [13] A.H. Zhuang, D.J. Valentino and A.W. Toga, Skull-stripping magnetic resonance brain images using a model-based level set, *Neuroimage* **32** (2006), 79–92.
- [14] K. Djemal, W. Puech, B. Rossetto, et al., Automatic active contours propagation in a sequence of medical images, *International Journal of Image and Graphics* **6** (2006), 267–292.
- [15] J. Ehrhardt, H. Handels and T. Malina, Atlas-based segmentation of bone structures to support the virtual planning of hip operations, *International Journal of Medical Informatics* **64** (2001), 439–447.
- [16] F. Yokota, T. Okada and M. Takao, Automated segmentation of the femur and pelvis from 3D CT data of diseased hip using hierarchical statistical shape model of joint structure, in: *MICCAI 2009*, G.Z. Yang, et al., Eds., Springer-Verlag, Berlin Heidelberg, 2009, pp. 811–818.
- [17] R.A. Zoroofi, Y. Sato and T. Sasama, Automated segmentation of acetabulum and femoral head from 3-D CT images, *IEEE Transactions on Information Technology in Biomedicine* **7** (2003), 329–341.
- [18] M. Sanderse, H.A. Marquering and E.A. Hendriks, Automatic initialization algorithm for carotid artery segmentation in CTA images, in: *Medical Image Computing and Computer Assisted Intervention*, J. Duncan and G. Gerig, Eds., Springer-Verlag Berlin Heidelberg, 2005, 846–853.
- [19] S. Milan, H. Vaclav and B. Roger, *Image Processing, Analysis, and Machine Vision*, 4th edi., Nelson Engineering, 2007.
- [20] L. He, Z.G Peng and E. Bryan, et al., A comparative study of deformable contour methods on medical image segmentation, *Image and Vision Computing* **26** (2008), 141–196.
- [21] C.M. Li, C.Y. Xu, C.F. Gui, et al., Distance regularized level set evolution and its application to image segmentation, *IEEE Transactions on Image Processing* **19** (2010), 3243–3254.
- [22] G. Aubert and P. Kornprobst, *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*, Springer-Verlag, New York, 2002.