Pattern Recognition Letters 31 (2010) 355–360

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Curvelet-based geodesic snakes for image segmentation with multiple objects

Hao Shan [a], Jianwei Ma [a,b,*]

[a] School of Aerospace, Tsinghua University, Beijing 100084, China
[b] Center of Geoscience, Ecole des Mines de Paris, 35 rue Saint-Honore, 77305 Fontainebleau, France

ARTICLE INFO

Article history:
Received 30 April 2009
Received in revised form 20 October 2009
Available online 28 October 2009

Communicated by G. Borgefors

Keywords:
Curvelets
Geometric snakes
Geodesic active contours
Image segmentation
Multiscale
Wavelets

ABSTRACT

Curvelet transform is a multiscale and multidirectional geometric wavelet transform, which is an optimal sparse representation of edges and contours of objects. In this paper, a curvelet-based geodesic snake (CGS) is proposed for image segmentation of multiple objects. By producing the edge map of objects by curvelet thresholding instead of simple gradient methods, the proposed method shows great promises to recognize edges of multiple objects with weak edges and strong noises. In addition, we design several rules to quantitatively compare the segmentation accuracy.

1. Introduction

Image segmentation is one of the focused topics in image processing and computer vision. Snake-based image segmentation has been widely applied in medical images (Yezzi et al., 1997), texture segmentation (Sagiv et al., 2006) and visual tracking (Paragios and Deriche, 2000; Niethammer et al., 2006).

Traditional snake mainly refers to the parametric snake (Kass et al., 1987; Xu and Prince, 1998a,b). The main limitations of the traditional snake are that it is dependent on parameters and cannot deal with topology changes. To solve the problems, geometric snake was proposed based on the mean curvature motion equation (Caselles et al., 1993; Malladi et al., 1994). However, the geometric snake uses heuristic stopping procedures when an edge is reached, and the image force cannot realize a complete stop. An improved version named geodesic active contour (GAC) was then proposed (Caselles et al., 1997). These two versions are both implemented by level sets (Osher and Sethian, 1988). The GAC is a particular case of the classical energy-based geometric snake, and equivalent to finding a curve of minimal weighted length in Riemannian space based on geodesic or local minimal distance computations. The GAC uses the negative gradient of the edge detector as an attraction force to project on the normal of the curve. This force balances the other terms close to the boundaries and causes the curve to stop there. In this paper, we follow the GAC method.

Improvements of active contours have been presented in the last few years. Paragios and Deriche (2002) presented a curve-based variational framework that integrates boundary- and region-based information modules to deal with partition problems. Chan and Vese (2001) proposed a model of image region segmentation where the stopping term at object boundaries does not depend on the image gradient. Xie and Mirmehdi (2004) proposed a region-aided geometric snake that integrates gradient flow forces with region constraints, thus more tolerant towards weak edges and noise. Xie and Mirmehdi (2008) also introduced a magnetostatic active contour model where the external force field is based on magnetostatics and hypothesized magnetic interactions between contours and object boundaries.

Applying multiscale methods to snakes is one of the hot topics in image segmentation. Coarse-to-fine scale strategies make the segmentation algorithms more economical for computation. Wu et al. (2000) proposed a directional image force for active contours based on wavelet frames. The wavelet-based snakes are helpful for noise suppression. Bresson et al. (2006) applied linear scale space into the parametric snake model. Mignotte and Meunier (2001) presented a multiscale approach for deformable contour optimization relying on a multigrid coarse-to-fine relaxation algorithm. Mukherjee and Acton (2002) introduced an integrated approach to cloud tracking based on scale-space classification. Keserci and Yoshida (2002) proposed an edge-guided wavelet snake model to fit multiscale edges. Liu and Hwang (2003) proposed an integrated wavelet-based snake model for segmentation and tracking based on the coarse-to-fine strategy. Tang and Acton (2004) combined

* Corresponding author. Address: School of Aerospace, Tsinghua University, Beijing 100084, China. Tel.: +86 10 62781824.
E-mail address: jma@tsinghua.edu.cn (J. Ma).

0167-8655/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2009.10.018

the B-spline model with multiscale gradient vector flow external force to elude clutter and to reliably localize the vessel boundaries. Recently, a curvelet-based parametric snake has been proposed by Ma et al. (2006) for multiscale tracking of geophysical fluids, which can deal with the segmentation of images with weak edges and noise in an efficient way. However, this curvelet-based snake is performed only for single-object detection and tracking.

In this paper, we extend the snake model proposed in (Ma et al., 2006) to multi-object detection and segmentation by combining the curvelets with the traditional GAC model.

2. Curvelets

Curvelet transform is a new multiscale geometric wavelet transform, which can represent edges and curve-singularities much more efficiently than traditional wavelet transforms. Curvelets can also capture intrinsic geometric structures of objects with fewer coefficients for a given accuracy of reconstruction in comparison with wavelets. The needle-shaped elements of curvelets are shown in Fig. 1. In this paper, we apply the second generation discrete curvelet transform (DCuT) (Candès and Donoho, 2004; Candès et al., 2006), which is implemented in four steps: (1) using 2-D fast Fourier transform for the image; (2) forming the product of the scale and angle windows; (3) wrapping the aforesaid product around the origin; and (4) applying the 2-D inverse fast Fourier transform.

For the triple index (j, l, k), where j, l and $k = (k_1, k_2)$ are scale, orientation, and translation parameters, respectively, curvelets are defined as functions of $x \in \mathbb{R}^2$ by

$$\phi_{j,l,k}(x) = \phi_j(R_{\theta_{j,l}}(x - x_k^{(j,l)})), \tag{1}$$

where $x = (x_1, x_2) \in \mathbb{R}^2$. Curvelets are scaled by 2^{-j}; $\theta_{j,l} = 2\pi l \cdot 2^{-\lfloor j/2 \rfloor}$, $0 \leqslant l \leqslant 2^{\lfloor j/2 \rfloor} - 1$ is an equispaced sequence of rotation angles, where $\lfloor \cdot \rfloor$ stands for integer part; $x_k^{(j,l)} = R_{\theta_{j,l}}^{-1}(k_1 2^{-j}, k_2 2^{-\lfloor j/2 \rfloor})^T$, $(k_1, k_2) \in \mathbb{Z}^2$; $R_{\theta_{j,l}}$ denotes the rotation matrix with angle $\theta_{j,l}$. Each function $y \in L^2(\mathbb{R}^2)$, can be represented as a curvelet series

$$y = \sum_{j,l,k} \langle y, \phi_{j,l,k} \rangle \phi_{j,l,k},$$

then curvelet coefficients are represented by

$$c_{j,l,k} = \langle y, \phi_{j,l,k} \rangle = \int_{\mathbb{R}^2} y(x) \phi_{j,l,k}(x) dx. \tag{2}$$

The fast curvelet transform algorithm is provided by Candès et al. (2006). Contrary to wavelets for which many choices of mother functions are available, only a few are expendable for the considered curvelets (they do not have an analytic representation). The design of curvelet mother functions is implemented by a technique of frequent windows, which should obey the admissibility conditions (Candès and Donoho, 2004). The currently used curvelets are based on the scaled Meyer window functions (Candès et al., 2006). For more details, one can refer to (Candès and Donoho, 2004; Candès et al., 2006) or to the recent reviewal papers (Ma and Plonka, 2009,).

3. Curvelet-based geometric snakes

The traditional snake is defined by an evolution course of a closed curve $\mathscr{C}(s) = [x(s), y(s)]^T$, $s \in [0,1]$ obeying a specific energy formula (Kass et al., 1987)

$$E(\mathscr{C}, \mathscr{C}_s, \mathscr{C}_{ss}) = \int_0^1 \left[\frac{1}{2} \alpha(s) |\mathscr{C}_s|^2 + \frac{1}{2} \beta(s) |\mathscr{C}_{ss}|^2 + g(\mathscr{C}) \right] ds, \tag{3}$$

where $\mathscr{C}_s$ and $\mathscr{C}_{ss}$ denote the first and second order partial derivatives of the curve $\mathscr{C}$ with respect to the parameter s, respectively. The $\alpha(s)$ and $\beta(s)$ are dependent on design parameters (usually set constants) of an internal energy function controlling the elasticity and rigidity of the snake. The last term in Eq. (3) represents an external energy. The basic idea is to force the closed curve to approach the object edges. This is also a method of seeking for the minimum of the energy, depending on image influences. The edge detector g is a positive and decreasing stopping function, which can be any decreasing function of the edge map f.

One can also preserve the smoothing of the curve with $\beta = 0$. Then Eq. (3) can be reduced to

$$E(\mathscr{C}, \mathscr{C}_s) = \int_0^1 \left[\frac{1}{2} \alpha(s) |\mathscr{C}_s|^2 + g(\mathscr{C}) \right] ds. \tag{4}$$

By minimizing E in Eq. (4) for $\mathscr{C}$, the geodesic active contours can be derived (Caselles et al., 1997). Without loss of generality, one can transform the minimization problem into a problem of geodesic computation in a Riemannian space (Caselles et al., 1997),

$$min \int_0^1 g(\mathscr{C}) |\mathscr{C}_s(s)| ds. \tag{5}$$

In order to minimize Eq. (5), one needs to search for its gradient descent process. One can compute the Euler–Lagrange equation as its gradient descent process (Caselles et al., 1997)

$$\frac{\partial \mathscr{C}}{\partial t} = (g\mathscr{K} - \langle \nabla g, \mathscr{N} \rangle)\mathscr{N}. \tag{6}$$

Here $\mathscr{N}$ is the unit inward normal to the curve and $\mathscr{K}$ is the curvature value. Geodesic active contours are represented implicitly as level sets of 2-D distance functions which evolve according to a Eulerian formulation. They are based on the theory of curve evolution and are implemented via level sets. Slightly different from the standard GAC, the constant term similar to the balloon force (Cohen, 1991) in the curve evolution equation is not considered in this paper. If the curve in Eq. (6) is embedded in a higher dimensional level set function Φ, the problem of solving the geodesic problem can be equivalent to searching for the steady state solution ($\partial\Phi/\partial t = 0$) of

$$\frac{\partial \Phi}{\partial t} = g\mathscr{K}|\nabla\Phi| + \nabla g \cdot \nabla\Phi, \quad \Phi(0, \mathscr{C}) = \Phi_0(\mathscr{C}), \tag{7}$$

where Φ_0 is the initial value of Φ when $t = 0$, and the curve $\mathscr{C}$ is considered as the zero level set of the level set function Φ. It is parameter free and can naturally handle topological changes. The unit normal $\mathscr{N}$ and the curvature $\mathscr{K}$ can be represented as functions of Φ, i.e., $\mathscr{N} = -\nabla\Phi/|\nabla\Phi|$ and $\mathscr{K} = \nabla \cdot (\nabla\Phi/|\nabla\Phi|)$.

Typically, the edge maps can be taken as $f = |\nabla I(u,v)|^2$ and $f = |\nabla G_\sigma(u,v) * I(u,v)|^2$ (Xu and Prince, 1998a,b) where $I(u,v)$ denotes the image intensity at position (u,v);$G_\sigma(u,v)$ is a 2-D Gaussian function with standard derivation σ and ∇ is the gradient operator. The multiscale edge map can be produced by a curvelet thresholding (Ma et al., 2006)

$$f = \sum_\gamma \tau(c_\gamma) \phi_\gamma, \tag{8}$$

in which τ is a thresholding function, γ is the collection of triple index (j,l,k) and $c_\gamma = \langle I, \phi_\gamma \rangle$ are the curvelet coefficients of the image. It is actually a denoising and edge detecting processing. Different from the strategy in (Ma et al., 2006), which reconstructs the coefficients to the original size of the image, we obtain the CGS's multiscale edge maps of each scale by reconstructing the coefficients at this scale. Suppose the maximal scale is denoted by j_0, the size of reconstructed images at each scale j has a $2^{j_0 - j}$ downsampling factor, in comparison with the original image size. Let f_{j_m} be the edge map at scale j_m, then we have

$$g_{j_m} = \frac{1}{1 + f_{j_m}}. \tag{9}$$

This process is repeated through the scale set $\zeta = [2, \ldots J]$, $j_m \in \zeta$, where J is the appointed maximal decomposed scale. Curve in Eq.

(7) can be formulated into the curvelet-based multiscale evolution equations using level sets

$$\frac{\partial \Phi_\zeta}{\partial t} = g_\zeta \mathcal{K}_\zeta |\nabla \Phi_\zeta| + \nabla g_\zeta \cdot \nabla \Phi_\zeta, \quad \zeta = [2, \ldots, J]. \tag{10}$$

Here Φ_ζ, g_ζ and $\mathcal{K}_\zeta$ denote respectively the level set functions, edge detectors and curvatures on scale set ζ.

In the coarsest scale, the size of coefficient subband is the smallest one and the contours should only sketch out the approximate figures of the objects. In other words, the convergence speed to the rough regions is fast, but the convergence accuracy is rough. In finer scales, the CGSs only need to evolve a few steps to converge by taking the results in the last scale as initial contours. An upsampling operator is used to realize the CGS transplant from a coarser scale to a finer scale to match the change of image sizes between the successive scales. Edge maps of the CGSs produced by curvelet thresholding take advantage of the properties of curvelet denoising and curve recognizing abilities. The segmentation becomes more and more accurate as scales increase. The multiscale strategy is also useful to reduce the computational cost (see e.g., Bresson et al., 2006). Algorithm of the CGS can be summarized as follows:

1. Initialization
   - 1.1 Perform the second generation DCuT of the original image to get the curvelet coefficients, and then thresholding.
   - 1.2 Reconstruct curvelet coefficients at the coarsest scale to get its edge map f_2.
   - 1.3 Initialize CGS at the coarsest scale and set a correct level set function Φ_2.
2. Evolving process For scales $j = 2, \ldots, 6$, implement the following steps:
   - 2.1 Evolve the level set function Φ_j at scale j using the edge maps f_j until convergence is reached.
   - 2.2 Get the converging zero level set (CGS contours) at scale j as the initial CGS of the next finer scale.
   - 2.3 If $j + 1 \leqslant 6$, go to the next finer scale $j + 1$, and go back to Step 2.1. If $j + 1 > 6$, stop.

Fig. 1. Elements of curvelets at different scales and directions.

4. Experimental results

We present numerical experiments by applying the CGS to various images with different shapes, topologies and noises. Firstly, we show the basic running process of the CGS and evaluate its performances in comparison with the standard GAC. Secondly, we

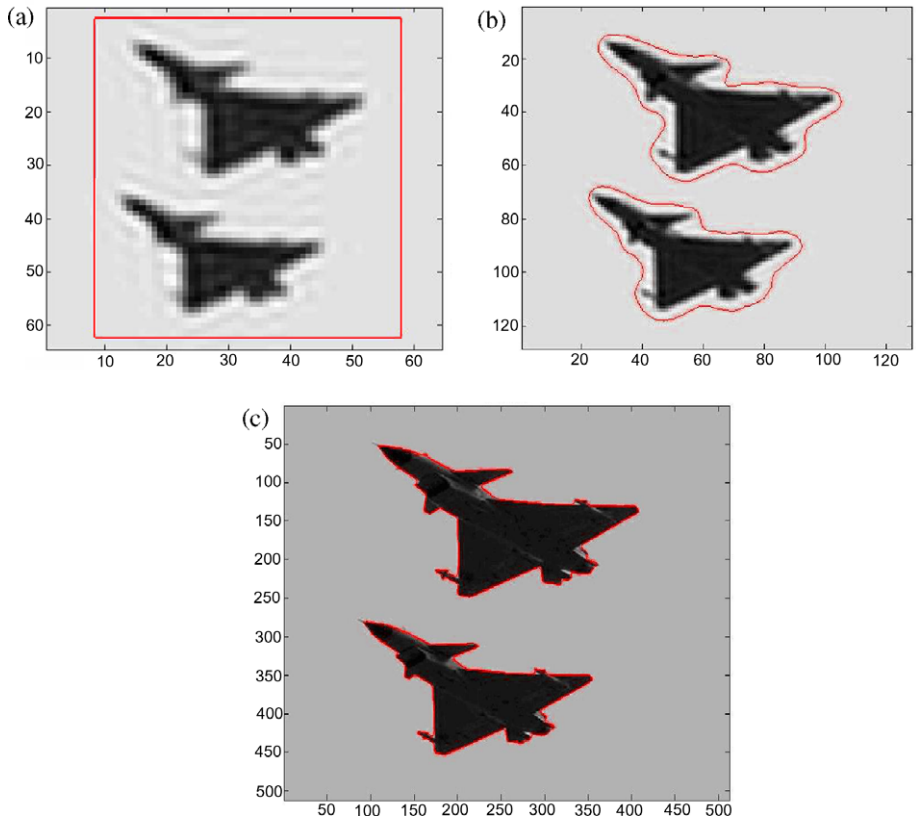

Fig. 2. Multiscale segmentation by the CGS. (a) Initial snake contour. (b) and (c) Results in two different scales.

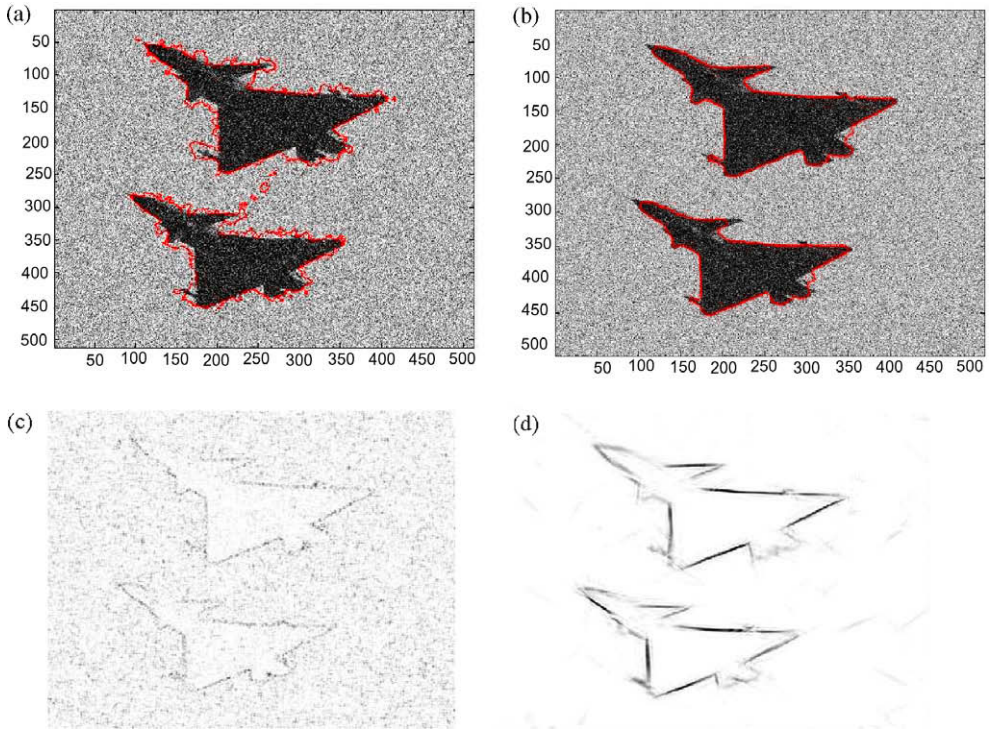

Fig. 3. Segmentation results (upper row) and edge maps (lower row). Left column: standard GAC. Right column: the proposed CGS.

show good ability of the CGS for segmentation of images with noise. Thirdly, a case with weak edges is tested.

4.1. Basic running process

In the first test, we exhibit the evolving results of the CGS on a 512×512 gray image. Fig. 2a shows an initial snake contour on the initial scale. Fig. 2b and c exhibit the evolving results in one coarse scale and the finest scale, respectively. The CGSs sketch out the rough outline of the objects sufficiently in a coarser scale. By taking the results (by scaling) as initial contours in the finer scale, the contours evolve closer to the true boundaries by using a few evolving steps. The CGSs achieve a good segmentation due to the application of the multiscale needle-shaped curvelet basis.

Table 1
Comparisons of the CPU durations (in s) and iteration step numbers between the CGS (all stages) and the standard GAC for the test shown in Fig. 3.

Algorithm phases of the CGS	Experimental results	
Curvelet composition	2.2903	
Curvelet reconstruction and Edge map generating (five scales)	47.1106	
	Time	Average single iteration time
Scale2 (iteration number)	0.1094(53 steps)	0.0021
Scale3 (iteration number)	0.2656(41 steps)	0.00648
Scale4 (iteration number)	1.5625(47 steps)	0.03324
Scale5 (iteration number)	12.5156(93 steps)	0.1346
Scale6 (iteration number)	6.1094(42 steps)	0.1455
Total time	69.9634(276 steps)	
Standard GAC:	Steps: 683;	Total time: 114.1563;
	Average time of single iteration: 0.1671	

4.2. Segmentation in noise environment

Fig. 3 shows the segmentation results obtained by the standard GAC and the ones obtained by the proposed CGS for a noisy image. The noise added to the original image is supposed to be zero-mean white additive Gaussian noise with the deviation 0.1. In Fig. 3a, it can be seen that the standard GACs almost fail to converge to the real edges because of their weak ability to deal with lots of local minima. Due to the convincing abilities of curvelets in noise removal and edge recognition, the CGSs well detect the objects. Smoothness of the final contours is also an important highlight of the robustness of the CGS in comparison to the standard method. Fig. 3c and d show the edge maps generated by the gradient method of the standard GAC and the CGS, respectively. The edge map in Fig. 3c is blurred and corrupted by noise, which leads to local maxima. The result in Fig. 3d benefits from the curvelet thresholding for noise removal.

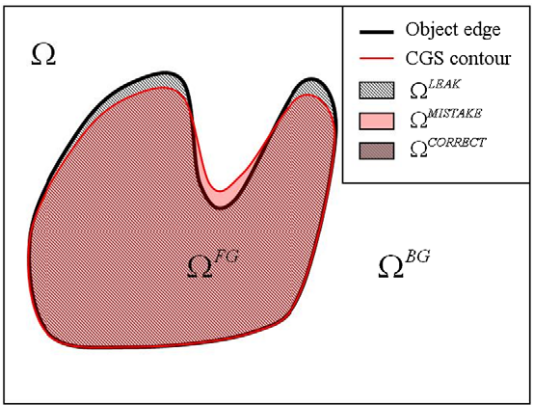

Fig. 4. Geometric relation of the quantitative segmentation measurement.

The computational time of the CGS is mainly related to the curvelet transform, to the image size and to the initial contours. In the test shown in Fig. 3, the CGS takes the total time of 69.96 s to implement the curvelet decomposition and reconstruction, to generate the edge map and to run the total 276 evolving steps. The standard GAC takes totally 114.1563 s and 683 evolving steps to achieve convergence results. The detailed CPU durations and iteration step numbers of the comparison between the CGS and the standard GAC are shown in Table 1.

Table 2
Comparisons of the quantitative segmentation accuracy measures (in %) between the CGS and the standard GAC for the test shown in Fig. 3.

Accuracy measures	The standard GAC (%)	The CGS (%)
MDR	17.3741	1.9139
LDR	0.1307	4.3782
OER	2.5855	0.9293

We design several rules to quantitatively compare the segmentation accuracy. Let Ω, Ω^{FG} and Ω^{BG} be the whole image region, the real foreground region of the objects, and the remaining background region, respectively. We have $\Omega = \Omega^{FG} \cup \Omega^{BG}$ (see Fig. 4). Let $\Omega^{MISTAKE}$ be the mistake detection, i.e., the regions in the background yet labelled "foreground region" by the CGS; Ω^{LEAK} be the leak detection, i.e., the regions in the foreground yet labelled "background region" by the CGS; and $\Omega^{CORRECT}$ be the correct detection, i.e., the regions in the foreground and labelled "foreground" by the CGS. The rules that we use to calculate the quantitative segmentation accuracy are described as follows: mistake detection rate $MDR = \Omega^{MISTAKE}/\Omega^{FG}$; leak detection rate: $LDR = \Omega^{LEAK}/\Omega^{FG}$; and the overall error rate $OER = (\Omega^{MISTAKE} + \Omega^{LEAK})/\Omega = (MDR + LDR) \times (\Omega^{FG}/\Omega)$, which measures the ratio of the mistake and leak pixels over the whole image pixels. The comparison values for the test in Fig. 3 are given in Table 2. Here the initial percentage of Ω^{FG}/Ω is 0.1477.

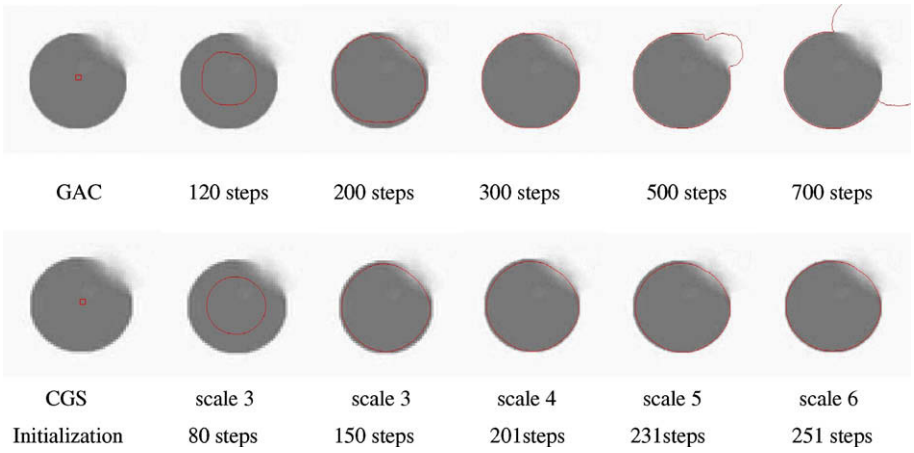

Fig. 5. The converging process of snakes for an image with weak edges. Upper row: standard GAC. Lower row: the proposed CGS.

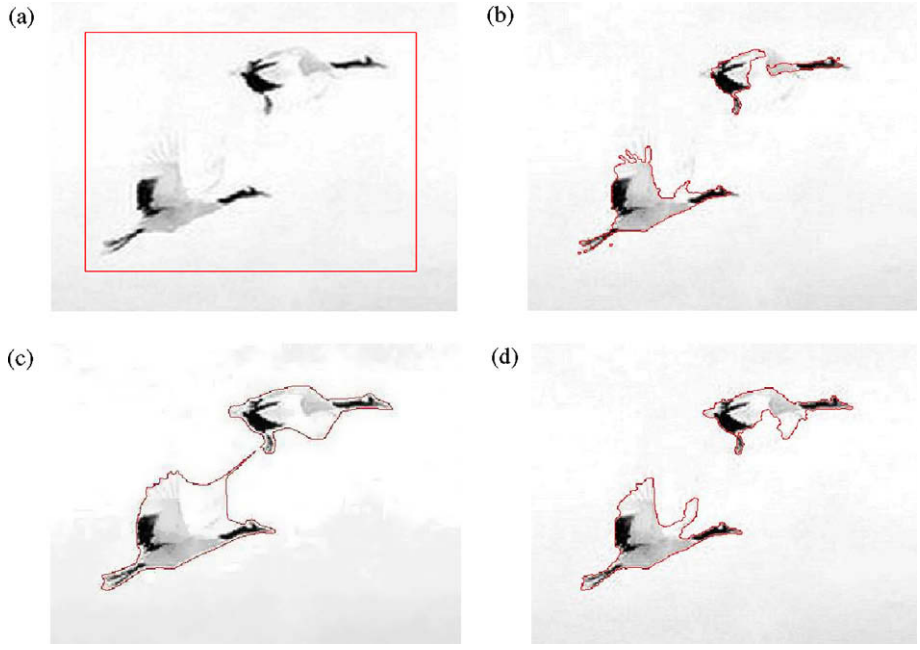

Fig. 6. Segmentation of multiple objects with weak edges. (a) Initial contour. (b) Final result of standard GAC. (c) Results by the CGS at curvelet scale 4. (d) Results by the CGS at curvelet scale 5 (final result).

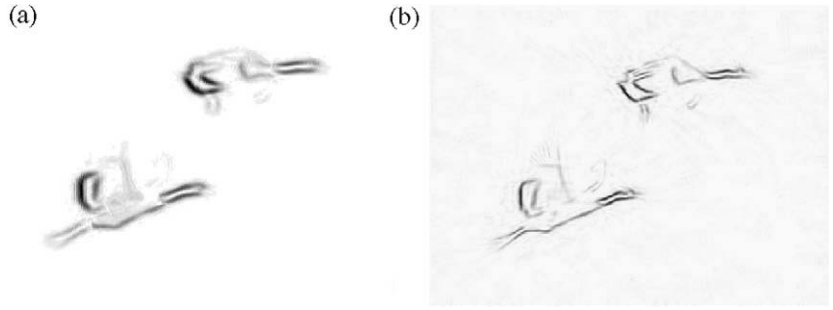

Fig. 7. Edge maps of the proposed CGS in segmenting the multiple objects with weak edges. (a) Edge map at curvelet scale 4. (b) Edge map at curvelet scale 5.

4.3. Segmentation with weak edges

As for the 256×256 disc-shaped pattern with an illegible area on the upper right edge, we also give a comparison between the standard GAC and the proposed CGS, as shown in Fig. 5. It can be seen that the CGS works well in terms of segmentation results and convergence speed. Moreover, being much more concentric to the outer disc contour than the standard GAC by visually inspecting, the CGS also exhibits a much more symmetrical and steady evolution.

In Fig. 6, we apply the proposed CGS to a 512×512 real image having multiple objects with weak edges and compare the results with the ones obtained by the GAC. The initial snakes for both methods are rectangles surrounding the two cranes. The cranes exhibit blurry edges located mainly at their wing feathers, and the upper crane has also a blurred abdomen. The CGSs can sketch out the figures of the crane and perform a steady and smooth process. Fig. 7 shows the edge maps at the two successive curvelet scales generated by the proposed CGS. The blurry edges located at the wing feathers are well recognized.

5. Conclusion

In this paper we propose a curvelet-based geometric snake for image segmentation with multiple objects. The main contribution of this paper is to extend the existent curvelet-based parameter snake for single-object segmentation to the geometric snake for multi-object segmentation. The edge map of the objects contained in the given images is obtained by curvelet thresholding, which, contrary to simple gradient methods, overcomes the issue of strong noise and weak edges. Curvelets are indeed well-suited for problems involving sharp discontinuities as edges contained in an image in the bidimensional case. Thus, in comparison with the traditional GAC, the proposed method is attractive since it can easily deal with noise background, and weak edges. Applications of the CGS to video tracking are our next work.

Acknowledgments

The authors thank the anonymous reviewers for their constructive comments and fruitful suggestions, which have contributed in improving the quality of the paper. The authors also thank financial support by National Natural Science Foundation of China under Grant Nos. 40704019 and 40674061, Tsinghua Basic Research Fund (JC2007030).

References

Bresson, X., Vandergheynst, P., Thiran, J., 2006. Multiscale active contours. Int. J. Comput. Vision 70 (3), 197–211.

Candès, E., Donoho, D., 2004. New tight frames of curvelets and optimal representations of objects with piecewise C^2 singularities. Commun. Pure Appl. Math. 57 (2), 219–266.

Candès, E., Demanet, L., Donoho, D., Ying, L., 2006. Fast discrete curvelet transforms. Multiscale Model. Simul. 5 (3), 861–899.

Caselles, V., Catté, F., Coll, T., Dibos, F., 1993. A geometric model for active contours in image processing. Numer. Math. 66, 1–31.

Caselles, V., Kimmel, R., Sapiro, G., 1997. Geodesic active contours. Internat. J. Comput. Vision 22 (1), 61–79.

Chan, T., Vese, L., 2001. Active contours without edges. IEEE Trans. Image Process. 10 (2), 266–277.

Cohen, L., 1991. On active contour models and balloons. Comput. Vision, Graph. Image Process.: Image Understanding 53 (2), 211–218.

Kass, M., Witkin, A., Terzopoulos, D., 1987. Snakes: Active contour models. Internat. J. Comput. Vision 1 (14), 321–331.

Keserci, B., Yoshida, H., 2002. Computerized detection of pulmonary nodules in chest radiographs based on morphological features and wavelet snake model. Med. Image Anal. 6 (4), 431–447.

Liu, J., Hwang, W., 2003. Active contour model using wavelet modulus for object segmentation and tracking in video sequences. Internat. J. Wavelets, Multiresolution Infor. Process. 1 (1), 93–112.

Ma, J., Plonka, G., 2009. Computing with curvelets: From image processing to turbulent flows. Comput. Sci. Eng. 11 (2), 72–80.

Ma, J., Plonka, G., in press. A review of curvelets and recent applications. IEEE Signal Process. Mag.

Ma, J., Antoniadis, A., Le Dimet, F.-X., 2006. Curvelet-based snake for multiscale detection and tracking of geophysical fluids. IEEE Trans. Geosci. Remote Sensing 44 (12), 3626–3638.

Malladi, R., Sethian, J., Vemuri, B., 1994. Evolutionary fronts for topology independent shape modeling and recovery. In: Proc. 3rd Eur. Conf. Computer Vision, pp. 3–13.

Mignotte, M., Meunier, J., 2001. A multiscale optimization approach for the dynamic contour-based boundary detection issue. Comput. Med. Imaging Graph. 25 (3), 265–275.

Mukherjee, D., Acton, S., 2002. Cloud tracking by scale space classification. IEEE Trans. Geosci. Remote Sensing 40 (2), 405–415.

Niethammer, M., Tannenbaum, A., Angenent, S., 2006. Dynamic active contours for visual tracking. IEEE Trans. Automatic Control 51, 562–579.

Osher, S., Sethian, J.A., 1988. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi Formulation. J. Comput. Phys. 79, 12–49.

Paragios, N., Deriche, R., 2000. Geodesic active contours and level sets for the detection and tracking of moving objects. IEEE Trans. Pattern Anal. Machine Intell. 22, 266–280.

Paragios, N., Deriche, R., 2002. Geodesic active regions: A new framework to deal with frame partition problems in computer vision. J. Vis. Commun. Image Represent 13 (1–2), 249–268.

Sagiv, C., Sochen, N., Zeevi, Y., 2006. Integrated active contours for texture segmentation. IEEE Trans. Image Process. 15, 1633–1646.

Tang, J., Acton, S., 2004. Vessel boundary tracking for intravital microscopy via multiscale gradient vector flow snakes. IEEE Trans. Biomed. Eng. 51 (2), 316–324.

Wu, H., Liu, J., Chui, C., 2000. A wavelet-frame based image force model for active contouring algorithms. IEEE Trans. Image Process. 9 (11), 1983–1988.

Xie, X., Mirmehdi, M., 2004. RAGS: Region-aided geometric snake. IEEE Trans. Image Process. 13 (5), 640–652.

Xie, X., Mirmehdi, M., 2008. MAC: Magnetostatic active contour model. IEEE Trans. Pattern Anal. Machine Intell. 30 (4), 632–646.

Xu, C., Prince, J.L., 1998a. Snakes, shapes, and gradient vector flow. IEEE Trans. Image Process. 7 (3), 359–369.

Xu, C., Prince, J.L., 1998b. Generalized gradient vector flow external forces for active contours. Signal Process. 71, 131–139.

Yezzi, A., Kichenassamy, S., Kumar, A., Olver, P., Annenbaum, A., 1997. A geometric snake model for segmentation of medical imagery. IEEE Trans. Med. Imaging 16, 199–209.