Fast hybrid fitting energy-based active contour model for target detection

Dengwei Wang (王登位), Tianxu Zhang (张天序), and Luxin Yan (颜露新)*

State Key Laboratory for Multispectral Information Processing Technologies,
Institute for Pattern Recognition and Artif icial Intelligence,
Huazhong University of Science and Technology, Wuhan 430074, China
**Corresponding author: yanluxin@gmail.com*

Received December 23, 2010; accepted March 3, 2011; posted online May 26, 2011

A novel hybrid fitting energy-based active contour model in the level set framework is proposed. The method fuses the region and boundary information of the target to achieve accurate and robust detection performance. A special extra term that penalizes the deviation of the level set function from a signed distance function is also included in our method. This term allows the time-consuming redistancing operation to be removed completely. Moreover, a fast unconditionally stable numerical scheme is introduced to solve the problem. Experimental results on real infrared images show that our method can improve target detection performance efficiently in terms of the number of iterations and the wasted central processing unit (CPU) time.

OCIS codes: 100.5010, 100.2960, 100.2000, 100.3008.
doi: 10.3788/COL201109.071001.

Since the introduction by Kass *et al.*[1], active contour ideas have been widely used in the computer vision field. Active contours can obtain smooth and closed contours to locate target boundaries with subpixel accuracy, a capability usually impossible in traditional methods[2,3]. Existing active contour methods can be roughly divided into two categories: edge-based methods[4,5] and region-based methods[6,7]. Edge-based methods use an image local gradient to drive the curve evolution process, which is usually sensitive to noise and weak edges commonly found in infrared imaging. On the other hand, instead of employing gradient information, region-based methods usually utilize certain statistical region information, such as intensity, color, and texture, to capture targets of interest. Both methods have advantages and disadvantages. When the imaging conditions are fine, the targets will appear with clear edges. In this case, the edge-based approach alone can achieve good detection results. However, when the opposite imaging conditions occur, utilizing the region-based method has more potential for success. Inspired by this phenomenon, in this letter, we propose a so-called hybrid fitting energy model to improve target detection performance utilizing both global image information and local image information.

We introduce the fitting term of Chan *et al.*[6] as our regional fitting term and formulate it in terms of level set function $\phi(x,y)$ as

$$\begin{aligned}\mathrm{Region}(\phi) =& \lambda_1 \cdot \iint_\Omega |\mathrm{Image}(x,y) - \mathrm{Mean_{in}}|^2 \\ & H[\phi(x,y)]\,\mathrm{d}x\mathrm{d}y \\ & + \lambda_2 \cdot \iint_\Omega |\mathrm{Image}(x,y) - \mathrm{Mean_{out}}|^2 \\ & \{1 - H[\phi(x,y)]\}\mathrm{d}x\mathrm{d}y, \end{aligned} \quad (1)$$

where λ_1 and λ_2 are positive constants, ϕ is the level set function with its zero level set corresponding to the evolution curve, $H(\phi)$ is the one-dimensional (1D) Heaviside function with $H(\phi) = 1$ if $\phi \geqslant 0$ or $H(\phi) = 0$ if $\phi < 0$, Image is the given image, and $\mathrm{Mean_{in}}$ and $\mathrm{Mean_{out}}$ are the intensity averages of Image(x,y) in $\phi \geqslant 0$ and $\phi < 0$, respectively.

Based on the geodesic active contour (GAC) model[4], we introduce the following weighted curve length as our edge fitting term:

$$\mathrm{Edge}(\phi) = \iint_\Omega g[|\nabla \mathrm{Image}(x,y)|] \cdot |\nabla H[\phi(x,y)]|\mathrm{d}x\mathrm{d}y, \quad (2)$$

where g is the boundary feature map related to the image gradient (g can be a decreasing function such as $g = \exp(-\eta|\nabla \mathrm{Image}|^2)$ or $g = 1/(1+\eta|\nabla \mathrm{Image}|^2)$ with η controlling the slope).

For a more accurate computation involving the level set function and its evolution, we need to regularize the level set function by penalizing its deviation from a signed distance function[8], which is characterized by

$$\mathrm{Penalize}(\phi) = \iint_\Omega \frac{1}{2}(|\nabla\phi| - 1)^2 \mathrm{d}x\mathrm{d}y. \quad (3)$$

As in many typical level set methods, we need to regularize the zero level set by penalizing its length to derive a smooth contour that is as short as possible during evolution:

$$\begin{aligned}\mathrm{Length}(\phi) &= \iint_\Omega |\nabla H\phi(x,y)|\mathrm{d}x\mathrm{d}y \\ &= \iint_\Omega \delta[\phi(x,y)]|\nabla\phi(x,y)|\mathrm{d}x\mathrm{d}y, \end{aligned} \quad (4)$$

where $\delta(\phi)$ is the 1D Dirac measure defined as $\delta(\phi) = \frac{\mathrm{d}}{\mathrm{d}\phi}H(\phi)$.

Hence, our hybrid fitting energy functional can be formulated as

$$\begin{aligned}\mathrm{Energy} = & \ \alpha \cdot \mathrm{Region}(\phi) + \beta \cdot \mathrm{Edge}(\phi) \\ & + \mu \cdot \mathrm{Penalize}(\phi) + \xi \cdot \mathrm{Length}(\phi), \end{aligned} \quad (5)$$

where α, β, μ, and ξ are weighting coefficients to balance the contribution of each term.

Keeping Mean_{in} and Mean_{out} fixed and minimizing the entire energy functional Energy with respect to ϕ, we deduce the associated Euler-Lagrange equation for ϕ as

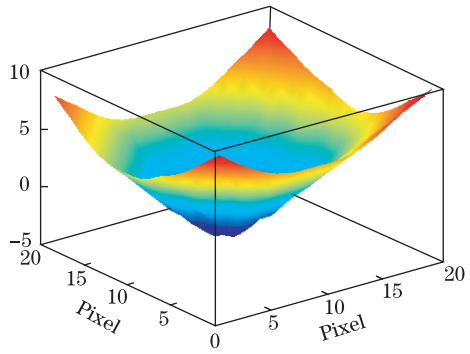
$$\begin{aligned}\frac{\partial \phi}{\partial t} &= \alpha \cdot \delta(\phi)[-\lambda_1(\text{Image} - \text{Mean}_{\text{in}})^2 \\ &+ \lambda_2(\text{Image} - \text{Mean}_{\text{out}})^2] \\ &+ \beta \cdot \text{div}\Big(g\frac{\nabla\phi}{|\nabla\phi|}\Big) + \mu \cdot \text{div}\Big[\Big(1 - \frac{1}{|\nabla\phi|}\Big)\nabla\phi\Big] \\ &+ \xi \cdot \delta(\phi)\text{div}\Big(\frac{\nabla\phi}{|\nabla\phi|}\Big). \end{aligned} \tag{6}$$

An explicit scheme is the most popular method for solving Eq. (6). However, due to the Courant-Friedreichs-Lewy (CFL)[9] condition, which asserts that numerical waves should propagate at least as fast as physical waves so that the curve can move only a small distance in each iteration, this requires a very small time step. Thus, if the curve is not near the edge of object of interest, the curve may take a long time to reach the final position. To remove the restriction on time step and obtain fast convergence, we introduce the fast additive operator splitting (AOS)[10] scheme to solve the terms marked by operator div in Eq. (6). The existence of $\delta(\phi)$ leads to some differences between our terms and the processing objects of AOS. Fortunately, Chan *et al.* indicated that $\delta_\varepsilon(\phi)$ could be replaced by $|\nabla\phi|$. Moreover, in the restriction on signed distance function, we have $|\nabla\phi| = 1$; thus, our equation will be an appropriate object that can be handled by AOS. The first term on the right side of Eq. (6) has no relation with the gradient of the level set function and can thus be treated as a constant. In our previous work, we gave the AOS scheme for the terms marked by operator div[11].

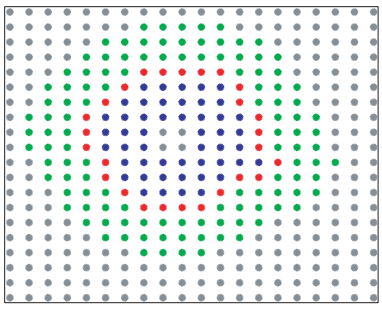
A direct implementation of Eq. (6) involves the re-estimation of the level set function at all pixels and not simply the zero level set corresponding to the current front. This front propagation method is computationally expensive because of the large amount of useless calculations that has to be performed for the pixels during the front propagation. In order to overcome this drawback, a narrow band approach which is initially proposed[12] and extensively analyzed and optimized[13] is proposed. The key idea is to deal only with the pixels close to the latest position of the zero level set in both directions (inwards and outwards). As the curve evolution is performed smoothly according to the Euler-Lagrange equation, the use of pixels far from the current contour does not affect the evolution. Therefore, only pixels close to the current contour are considered. A set of narrow band pixels is defined around the current contour and the level set function is updated only within this band.

9.2195	8.4853	7.8102	7.2111	6.7082	6.3246	6.0828	6	6	6	6	6.0828	6.3246	6.7082	7.2111	7.8102	8.4853	8.9443	9.4340	10
8.4853	7.8102	7.0711	6.4031	5.8310	5.3852	5.0990	5	5	5	5	5.0990	5.3852	5.8310	6.4031	7.0711	7.6158	8.0623	8.6023	9.2195
7.8102	7.0711	6.4031	5.6569	5	4.4721	4.1231	4	4	4	4	4.1231	4.4721	5	5.6569	6.3246	6.7082	7.2111	7.8102	8.4853
7.0711	6.4031	5.6569	5	4.2426	3.6056	3.1623	3	3	3	3	3.1623	3.6056	4.2426	5	5.3852	5.8310	6.4031	7.0711	7.8102
6.4031	5.6569	5	4.2426	3.6056	2.8284	2.2361	2	2	2	2	2.2361	2.8284	3.6056	4.1231	4.4721	5	5.6569	6.4031	7.0711
5.8310	5	4.2426	3.6056	2.8284	2.2361	1.4142	1	1	1	1	1.4142	2.2361	2.8284	3.1623	3.6056	4.2426	5	5.6569	6.4031
5.3852	4.4721	3.6056	2.8284	2.2361	1.4142	1	0	0	0	0	1	1.4142	2	2.2361	2.8284	3.6056	4.2426	5	5.8310
5	4.1231	3.1623	2.2361	1.4142	1	0	−1	−1	−1	−1	0	1	1	1.4142	2.2361	2.8284	3.6056	4.4721	5.3852
4.4721	3.6056	2.8284	2	1	0	−1	-1.4142	−2	−2	-1.4142	−1	0	0	1	1.4142	2.2361	3.1623	4.1231	5.0990
4.1231	3.1623	2.2361	1.4142	1	0	−1	−2	-2.8284	-2.8284	-2.2361	-1.4142	−1	−1	0	1	2	3	4	5
4	3	2	1	0	−1	-1.4142	-2.2361	-3.1623	-3.6056	-2.8284	−2	−1	0	1	1.4142	2.2361	3.1623	4.1231	5.0990
4	3	2	1	0	−1	−2	-2.8284	-3.6056	-3.6056	-2.8284	−2	−1	0	1	2	2.8284	3.6056	4.4721	5.3852
4	3	2	1	0	−1	-1.4142	-2.2361	-2.8284	−3	-2.2361	-1.4142	−1	0	1	2	3	4	5	5.8310
4.1231	3.1623	2.2361	1.4142	1	0	−1	-1.4142	−2	−2	−2	−1	0	1	1.4142	2.2361	3.1623	4.1231	5.0990	6.0828
4.4721	3.6056	2.8284	2.2361	1.4142	1	0	−1	−1	−1	−1	−1	0	1	2	2.8284	3.6056	4.4721	5.3852	6.3246
5	4.2426	3.6056	2.8284	2.2361	1.4142	1	0	0	0	0	0	1	1.4142	2.2361	3.1623	4.1231	5	5.8310	6.7082
5.6569	5	4.2426	3.6056	2.8284	2.2361	1.4142	1	1	1	1	1	1.4142	2.2361	2.8284	3.6056	4.4721	5.3852	6.3246	7.2111
6.4031	5.6569	5	4.2426	3.6056	2.8284	2.2361	2	2	2	2	2	2.2361	2.8284	3.6056	4.2426	5	5.8310	6.7082	7.6158
7.0711	6.4031	5.6569	5	4.2426	3.6056	3.1623	3	3	3	3	3	3.1623	3.6056	4.2426	5	5.6569	6.4031	7.2111	8.0623
7.8102	7.0711	6.4031	5.6569	5	4.4721	4.1231	4	4	4	4	4	4.1231	4.4721	5	5.6569	6.4031	7.0711	7.8102	8.6023

(a)



(b)



(c)

Fig. 1. (Color online) Narrow band example. (a) A level set function (signed distance function) matrix, (b) 3D expression of (a), and (c) narrow band region.

The narrow band algorithm we employ in this letter to update our level set function consists of the following steps:

(1) Check whether the current level set function is a signed distance function. If it is a signed distance function, then go to the next step; otherwise, stop the execution.

(2) Set the half-width parameter K of the narrow band and search for the coordinates satisfying the condition $-K \leqslant \mathrm{LSF} \leqslant K$. These coordinates make up our narrow band pixels.

(3) According to the specified Euler-Lagrange equation, update the level set function in a small range of narrow band pixels.

(4) When the level set function changes, repeat steps (1) to (3).

(5) When the convergence condition is met, end the iteration process.

In this letter, we take a level set function as our example to explain the narrow band approach, as shown in Fig. 1. Figure 1(a) is a level set function (signed distance function with size equal to 20×20 pixels) matrix, (b) the three-dimensional (3D) expression of (a), and (c) the narrow band region. The blue pixels denote the inner points of the narrow band, the green pixels the outer points, and the red pixels denote the zero level sets corresponding to the current level set function, which together constitute the current narrow band pixels. When we update the current level set function, we only need to deal with the narrow band pixel values while the remaining positions (pixels shown in gray) remain unchanged.

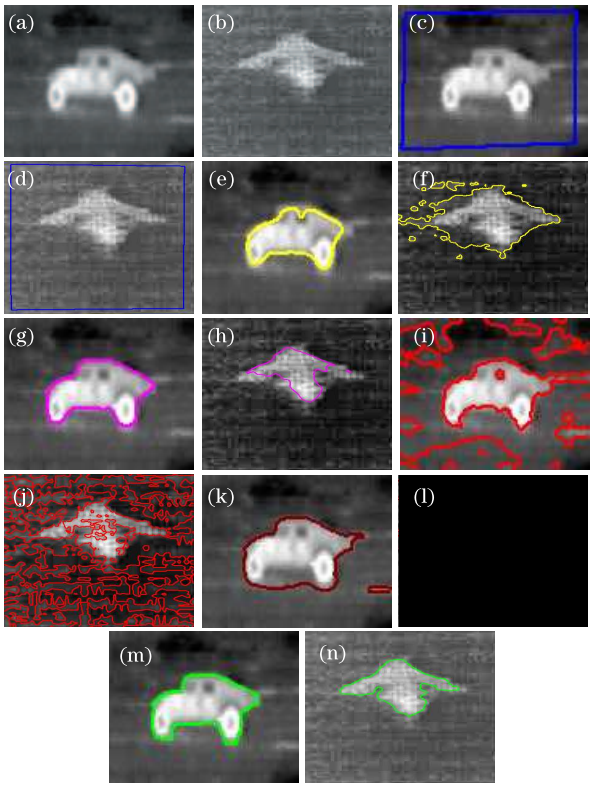


Fig. 2. Comparison of our model with four other models on two real infrared images. The controlling parameters of our model are selected as: $\alpha = 1$, $\beta = 30$, $\mu = 1$, and $\xi = 0.00001 \times 255 \times 255$.

Then we evaluate the performance of our method on real infrared target images. The experiments are implemented through Matlab R2008a on a computer with Intel Core 2 Duo 2-GHz central processing unit (CPU), 2-G random access nemory (RAM), and Windows XP operating system. In the following experiments, we use some default settings for partial parameters: $\lambda_1 = 1$, $\lambda_2 = 1$, $\Delta t = 0.1$ (the time step). Only the length parameter, ξ, which has a scaling role, is not the same in all experiments. If we have to detect all or as many targets as possible and of any size, then ξ should be small. If we have to detect only larger targets and not detect smaller objects, then ξ has to be larger. We thus give the exact value of ξ each time, together with the controlling parameters α, β, and μ.

Figure 2 compares the detection performance of the Chan-Vese model, the GAC model, the method by Zhang *et al.*[14,15], and our model on two real infrared images, which include some rather weak boundaries. Moreover, significant intensity variations also exist in these infrared images. Figures 2(a) and (b) are the original input images, the sizes of which are 118×93 and 310×316 pixels, respectively. Figures 2(c) and (d) show the initial curves. Figures 2(e) and (f) show the results by the Chan-Vese model. An over-detection phenomenon at the upper part of the image can be seen from Fig. 2(e), whereas Fig. 2(f) includes some false alarms. Figures 2(g) and (h) show the results by the GAC model. We can see from Fig. 2(h) that the evolution process ignores the weak region directly because this model relies on local gradient information alone. Figures 2(i) and (j) show the results using the method of Zhang *et al.*[14,15]. The results from the two methods indicate a mass of false alarms. Figures 2(m) and (n) show the results from our method. The results shown in Figs. 2(g) and (m) are almost the same. However, a closer inspection of the results indicates a distinction between the proposed model and the GAC model. Taking the bottom of the right wheel as an example, we can see that the GAC model does not detect the fuzzy part of the bottom of the right wheel, yet our approach achieves very good results. We can clearly see an improvement from the zoomed view of Fig. 3. Thus, compared with the other methods mentioned here, our method obtains the best detection result because our approach takes into account the edge and region information of the image simultaneously.

Quantitatively, we use a popular measure called the Dice coefficient[16] to compare the final detection results obtained by the four methods. Given two target regions Ω_1 and Ω_2 from two different algorithms, the Dice coefficient is defined as

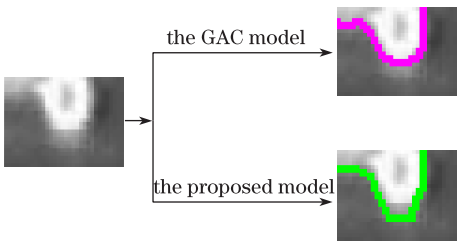


Fig. 3. Zoomed view of the fuzzy part of the bottom of the right wheel.

Table 1. Performance Comparison of Several Detection Methods

Detection Methods	Chan-Vese Model		GA Model		Method of Zhang *et al.*[14]		Method of Zhang *et al.*[15]		the Proposed Method	
Metrics	Fig. 2(a)	Fig. 2(b)	Fig. 2(a)	Fig. 2(b)	Fig. 2(a)	Fig. 2(b)	Fig. 2(a)	Fig. 2(b)	Fig. 2(a)	Fig. 2(b)
d_{v}	0.8723	0.6570	0.9345	0.8378	0.5285	0.1451	0.8605	0.9079	0.9905	0.9693
Number of Iterations	82	250	30	85	35	50	70	200	11	18
Total CPU Times (s)	45.1902	137.7854	65.9280	186.7925	11.2893	62.5065	17.6355	98.6552	0.8911	4.3172

$$d_{\mathrm{v}}(\Omega_1,\Omega_2)=\frac{2\mathrm{Area}(\Omega_1\cap\Omega_2)}{\mathrm{Area}(\Omega_1)+\mathrm{Area}(\Omega_2)}. \quad (7)$$

The Dice coefficient varies from 0 to 1, and it measures the degree of agreement between the two detected regions. It is 1 when the two regions are identical and 0 when they are completely different. We record the Dice coefficients of the five different methods in Table 1.

In addition, to demonstrate the superiority of our method in terms of evolution velocity, we provide the number of iterations and total CPU time (unit: s) for each of the methods described in Fig. 2, as shown in Table 1. Based on Table 1, our method achieves the fastest evolution process (with the smallest CPU time) and the highest d_{v} value. Thus, our model clearly achieves the most accurate detection results.

Figuer 4 presents the results for another set of real-world images. The first row shows the result for a B29 plane image. The contour is placed across the two planes. For this image, we use the parameters $\alpha=1$, $\beta=10$, $\mu=1$, and $\xi=0.000125\times255\times255$. The image in the second row is corrupted by intensity homogeneity due to nonuniform infrared thermal radiation, which is often seen in the infrared imaging of a sea surface target. New contours can emerge during the evolution to extract multiple object boundaries. For this image, we set $\alpha=1$, $\beta=11$, $\mu=1$, and $\xi=0.000016\times255\times255$ as the parameters. The third row shows the result for a dark ship target under sea background. The fact that the dark ship target is surrounded by the highly cluttered background with numerous sun-glints can be seen clearly. This makes it difficult to recover the whole object boundary if it relies on local gradient or global image information alone. Nevertheless, our method successfully extracts the object boundaries. In this case, we choose $\alpha=1$, $\beta=5$, $\mu=1$, and $\xi=0.000003\times255\times255$ as the parameters.

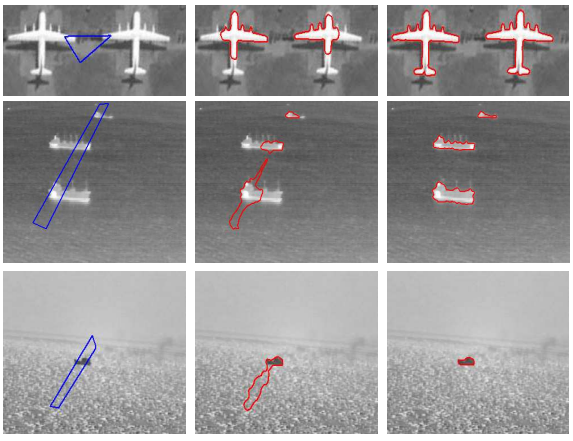


Fig. 4. Results of our method for real infrared images. The curve evolution process from the initial contour (in the first column) to the final contour (in the third column) is shown in every row for the corresponding image. The CPU time for the first row is 0.8073 s; 2.3168 s for the second row; and 2.1104 s for the third row.

In conclusion, a hybrid model that integrates the region and edge information while solving the efficient AOS scheme is presented. The experimental results show that the proposed method can improve detection performance effectively in terms of the number of iterations and the wasted CPU time.

This work was supported by the National Natural Science Foundation of China under Grant No. 60736010.

References

1. M. Kass, A. Witkin, and D. Terzopoulos, Int. J. Comput. Vis. **1,** 321 (1988).
2. X. Wang, L. Liu, and Z. Tang, Chin. Opt. Lett. **7,** 931 (2009).
3. H. Deng, J. Liu, and Z. Chen, Chin. Opt. Lett. **8,** 24 (2010).
4. V. Caselles, R. Kimmel, and G. Sapiro, Int. J. Comput. Vis. **22,** 61 (1997).
5. C. Xu and J. L. Prince, IEEE Trans. Image Process. **7,** 359 (1998).
6. T. F. Chan and L. A. Vese, IEEE Trans. Image Process. **10,** 266 (2001).
7. R. Ronfard, Int. J. Comput. Vis. **13,** 229 (1994).
8. C. Li, C. Xu, C. Gui, and M. D. Fox, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* 430 (2005).
9. J. A. Sethian, *Level Set methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science* (Cambridge University Press, New York, 1999).
10. J. Weickert, B. M. ter Haar Romeny, and M. A. Viergever, IEEE Trans. Image Process. **7,** 398 (1998).
11. D. Wang, T. Zhang, W. Shi, Z. Wang, X. Yang, and L. Wei, Opt. Eng. **49,** 037004 (2010).
12. D. Chop, J. Comput. Phys. **106,** 77 (1993).
13. D. Adalsteinsson and J. Sethian, J. Comput. Phys. **118,** 269 (1995).
14. K. Zhang, H. Song, and L. Zhang, Pattern Recogn. **43,** 1199 (2010).
15. K. Zhang, L. Zhang, H. Song, and W. Zhou, Image Vis. Comput. **28,** 668 (2010).
16. L. Dice, Ecol. **26,** 297 (1945).