Texture Image Segmentation Based on Nonlinear Diffusion

ZHANG Yu

Abstract  A texture image segmentation based on nonlinear diffusion is presented. The scale of texture can be measured during the process of nonlinear diffusion. A smooth 5-channel vector image with edge preserved, which is composed of intensity, scale and orientation of texture image, can be achieved by coupled nonlinear diffusion. A multi-channel statistical region active contour is employed to segment this vector image. The method can be seen as a kind of unsupervised segmentation because parameters are not sensitive to different texture images. Experimental results show its high efficiency in the semi-automatic extraction of texture image.

Keywords  texture image segmentation; nonlinear diffusion; statistical region active contour

CLC number  P237.4; TP753

Introduction

The goal of image segmentation is to divide an image into distinct homogeneous regions according to a certain criterion. Description and discrimination of image texture are important and are widely applied in image segmentations. Common methods of segmentation are statistical-based or filter-based, such as spatial grey level dependence matrices, Markov random fields, wavelet and Gabor transformation, etc. These methods have achieved a certain performance. However, they suffer from drawbacks of high computational cost and parameters choice. In many practical image processing tasks, segmentation can be greatly simplified by semi-automatic methods, which are convenient for combining segmentation with the operator’s knowledge and experience, such as medical image processing, man-made objects semiautomatic extraction in remote sensing, and so on. Usually, in such instances, we only need to get the region of interest (ROI) from its background instead of segmenting the entire image. Snake or active contour is a popular approach to get the boundaries of ROI. These ROIs play an important role in the automatic analysis of image contents. A snake or active contour is defined as an energy function to find the best match between a polygon and the boundary of the ROI. Lots of works in segmentation by active contour follow two basic rules: edge-based and region-based. An edge-based active contour is attracted by local image structure with high gradient response or high response of edge point detectors. Though the edge-based method is well explored, due to its sensitivity to noise and easy to be trapped in a false local minimum, it is extended by region-based cues. Region-based active contour uses region information as an external force to drive the snake to match the boundaries of an ROI. Such region information is composed of intensity, textures or colors of the region. The underlying homogeneity assumption is that the features of all pixels within one region are statisti-
cally independently distributed according to the same probability density function (PDF).

As a pre-processing technique, nonlinear diffusion can remove noise effectively while keeping the sharp edges. During the process of non-linearity diffusion, change of pixel value can be used for extracting the scale of the texture. By fusing the information on magnitude, scale, and structure tensor, a multi-channel vector image is generated to be utilized for texture discrimination. In this study, we suggest a strategy to segment the ROI in texture image by a 5-channel vector image. It is a semi-automatic approach based on statistical region active contour. Because of the insensitivity of parameter choice, it can be seen as one of the effective semi-automatic unsupervised texture segmentation.

1 Nonlinear diffusion and texture features extraction

Nonlinear diffusion is one of the selective smooth methods. In fact, it is an expansion of a Gaussian smooth and based on Perona and Malik’s works[1]. Its equation is:

$$\partial_t u = \text{div}(g(|\nabla u|)\nabla u)$$  \hspace{1cm} (1)

where $u(t=0)$ is the source image and $g$ is diffusion function. The $g$ function is:

$$g(|\nabla u|) = \frac{1}{|\nabla u| + \epsilon}$$  \hspace{1cm} (2)

with a small constant $\epsilon(0.0001)$ to avoid instability of the $g$ function.

From Eqs. (1) and (2), when inputting the source image, smaller gradient pixels are smoothed first, which yield small segmentation-like blocks. With $t$ increasing, these blocks melt together and become larger blocks. This allows for an efficient computation of texture scale because of the following proven rules in one dimension[2]:

1) $M$ neighboring pixels with the same value can be considered as a block with mass $m$;
2) during evolution, neighboring blocks melt together and become larger blocks;
3) blocks with extremum adapt their value to that of their neighbors with speed $2/m$;
4) boundary blocks adapt their value with speed $1/m$;
5) all other blocks keep their value unchanged.

The explanation of the above conclusion with Fig.1 are five blocks of pixels and their mass are $m_a, m_b, m_c, m_d$, and $m_e$, respectively. Block $a$ and $e$ are boundary blocks. Block $b$ is a local minimum block. Block $c$ is a local maximum block. Directions of their value change are shown as arrows in Fig.1. As mentioned above, the change speed of a block is inversely proportional to its mass. In Fig.1, block $d$ is a non-local extremum block; therefore, its value stays unchanged before merging with its neighbors. It has an opportunity to become an extremum block as long as it merges with others and thereby form a larger block. In 2 dimensions, the above rules are no longer exactly satisfied, but also have the similar result. Therefore, we can get the following conclusion: in the evolution of image $u$, the change speed of the pixel, which measures the scale of texture, approximately determines the size of block it belongs to. While solving Eq.(1), the relative scale is achieved by:

$$s = \frac{1}{4T} \int_0^T |\partial_t u|\,dt$$  \hspace{1cm} (3)

where $T$ is the number of iterations; $\int_0^T \delta(\partial_t u,0)\,dt$ is the number of iterations in which the pixel stays unchanged. The denominator of Eq.(3) is the change number of a pixel in all iterations. In fact, measuring the scale of the image by Eq.(3) is not accurate, but it is enough for us to discriminate the ROI from the texture image by a semi-automatic scheme.

![Evolution rules of pixels in 1D](image)

Apart from the scale of the texture, their orientation and magnitude are important features to discriminate textures. These features can be obtained from the structure tensor[3]:

$$J = G \ast \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix}$$  \hspace{1cm} (4)
where $I$ is the grayscale of a pixel and $G$ is a Gaussian window. Like the method introduced by Reference [4], we get three elements: $I_x^2, I_y^2, I_x I_y$, which contain information of texture orientation and magnitude. A 5-channel vector image $U(u_1, u_2, u_3, u_4, u_5)$ is generated by the union of $I_x^2, I_y^2, I_x I_y$ and $s$. We set $u_1 = I_x, u_2 = s, u_3 = I_y^2, u_4 = I_x^2, u_5 = I_x I_y$ as initial value of $U$, then, the process of 5-channel coupled nonlinear diffusion to the vector image $U$ is defined as:

$$\partial_t u_i = \text{div}(g(\sum_{k=1}^{N} |\nabla u_i|^2) \nabla u_i) \quad (i = 1, \ldots, N)$$

(5)

where $N = 5$; $g$ is a function and defined in Eq.(2). Compared to the Gaussian function, the nonlinear diffusion can smooth the vector image $U$ while keeping all edges sharp, so the boundary of ROI can be localized accurately. In this study, a semi-implicit additive operator splitting scheme (AOS) [5] is employed, which allows efficient computation of nonlinear diffusion. The AOS scheme is about several dozens of times faster than a simple explicit scheme while keeping errors imperceptible.

2 Semiautomatic region segmentation based on five channels

Active contour (or snake) is a classical model and frequently used in semiautomatic segmentation. In this study, a statistical region active contour is employed to extract ROI in a texture image. As mentioned above, the segmentation will fail if we carry out segmentation directly without considering the features of the texture. We make the following suppositions:

1) the distribution of pixels in any channel of the vector image is Gaussian;
2) it is not correlative between each channel;
3) the ROI is a single object to be extracted in the texture image.

We can simplify the computation and get better segmentation results by these suppositions. From supposition 1), the energy function of the statistical region active contour is:

$$E = N_a \log \sigma_a + N_b \log \sigma_b$$

(6)

where $a$ and $b$ are inner and outer regions (defined by active contour respectively) when the contour evolves; $N_a$ and $N_b$ are the number of pixels while $\sigma_a$ and $\sigma_b$ are standard deviations in these regions, respectively. The energy $E$ will achieve the minimum when the active contour gets the right segmentation. The key of the approach is that we can obtain fast algorithms by transforming the summations over a region, for the calculation of the statistics, into summations along the boundary of the region. Details of the algorithms are shown in Reference [6]. In view of vector image, using the above suppositions, the formula is modified as:

$$E' = N_a \sum_{i=1}^{N} \log \sigma_a^i + N_b \sum_{i=1}^{N} \log \sigma_b^i$$

(7)

Compared to Eq.(6), here, $N = 5$ is the number of channels in the vector image; $\sigma_a^i$ and $\sigma_b^i$ are standard deviations of two regions respectively in each channel. In Reference [8], the region $b$ is the entire image except the region $a$. To fit the operation of the semiautomatic method, we separate regions $a$ and $b$ from the background by manually setting the initial contour. As shown in Fig.2, the rectangular region is the entire image, the gray polygonal region is the ROI, the dashed line is the contour initialized manually and the solid is the current contour forced by Eq.(7). Therefore, the active contour should evolve in the region of the dashed line instead of the entire image and the region $b$ becomes $b'$ as illustrated in Fig.2. This treatment will bring the following advantages:

![Fig.2 Illustration of region reduction](image)

1) improve the efficiency greatly by removing a majority of the image;
2) avoid interferential objects in the image by manual techniques.

Fig.3 is a flowchart of the semiautomatic segmentation for texture images based on coupled nonlinear diffusion.
3 Experiment analysis and conclusions

Synthetic texture images, natural images and aerial images are employed to test the proposed approach. Parameters that need to be set during experiments are: iteration times $T (=30)$ and the AOS step (=1) in Eqs.(1) and (3); iteration times $T (=20)$ and the AOS step (=2) in Eq.(5). They are insensitive to different texture images. We keep them invariable in all experiments. Therefore, the proposed approach belongs to one of non-surveillance segmentation.

Fig.4 shows the results of the synthetic texture image segmentation. The ROI, which is in the middle of the image, has the same texture structure as the background. They only differ in scale and orientation. Fig.4(a) is the correct result of the 5-channel vector image segmentation. Fig.4(b) shows the wrong segmentation which is only based on the intensity of the texture image. Because of the same intensity between ROI and the background, the segmentation leads to failure inevitably. The scale, $I_x^2, I_y^2$ and $I_x I_y$ of each channel image, which is achieved by coupled nonlinear diffusion filter, are illustrated in Fig.4(c), 4(d), 4(e) and 4(f) respectively. Fig.5 shows the segmentation of aerial and natural texture images. The first row (Fig.5(a)-5(d)) is the result by coupled 5-channel nonlinear diffusion whereas the second row (Fig.5(e)-5(h)) is the result of segmentation only based...
on the intensity of the image. Initial contours are illustrated as exterior polygons in images of the first row. Fig.5(a) is a part of the aerial image. Its ROI has a special orientation, so the orientation of the texture is the key channel to ROI discrimination. Another part of this aerial image is shown in Fig.5(b), which has an ROI with prominent orientation and intensity, therefore, orientation and intensity become main roles in ROI extraction. The result in Fig.5(b) is obviously more favorable than that in Fig.5(f). Fig.5(c) and Fig.5(d) indicate the significance of scale in the natural image segmentation. Usually, the difference in the ROI intensity of the natural image is tremendous and that makes ROI extraction impossible when segmenting ROI merely based on intensity of the image. However, the scale between ROI and the background is different due to the characteristic of the optical sensor. Experiment results in Fig.5(c) and Fig.5(d) illustrated good performance of extraction by the 5-channel coupled nonlinear diffusion scheme.

Experiment results indicate that the scheme based on 5-channel coupled nonlinear diffusion to segment or extract ROI is much better suited for semiautomatic segmentation. Though the description of the texture by 5 channels is not very accurate, it can deal with most tasks of texture image segmentation with high efficiency.

References

[1] Perona P, Malik J(1990) Scale space and edge detection using anisotropic diffusion[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 12:629-639

[2] Brox T, Welk M, Steidl G, et al.(2003) Equivalence results for TV diffusion and TV regularization[M]// Griffin L D, Lillholm M. Scale-space methods in computer vision. Berlin: Springer-Verlag

[3] Rao A R, Schunck B G(1991) Computing oriented texture fields[J]. CVGIP: Graphical Models and Image Processing, 53:157-185

[4] Rousson M, Brox T, Deriche R(2003) Active unsupervised texture segmentation on a diffusion based feature space[C]. 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Madison, WI

[5] Weickert J, ter Haar Romeny B M, Viergever M A(1998) Efficient and reliable schemes for nonlinear diffusion filtering[J]. IEEE Transactions on Image Processing, 7(3):398-401

[6] Chesnaut C, Refregier P, Boulet V(1999) Statistical region snake-based segmentation adapted to different physical noise models[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 21(11):1 145-1 157