

# On using an analogy to heat flow for shape extraction

Cem Direkoğlu · Mark S. Nixon

Received: 5 June 2010 / Accepted: 24 May 2011 / Published online: 18 June 2011  
© Springer-Verlag London Limited 2011

**Abstract** We introduce a novel evolution-based segmentation algorithm which uses the heat flow analogy to gain practical advantage. The proposed algorithm consists of two parts. In the first part, we represent a particular heat conduction problem in the image domain to roughly segment the region of interest. Then we use geometric heat flow to complete the segmentation, by smoothing extracted boundaries and removing noise inside the prior segmented region. The proposed algorithm is compared with active contour models and is tested on synthetic and medical images. Experimental results indicate that our approach works well in noisy conditions without pre-processing. It can detect multiple objects simultaneously. It is also computationally more efficient and easier to control and implement in comparison with active contour models.

**Keywords** Feature extraction · Shape extraction · Segmentation · Heat flow

## 1 Originality and contribution

In this paper, we introduce a novel evolution-based image segmentation algorithm based on the heat flow analogy. Our model is new and different because of the evolution technique and organization of segmentation. The evolution is achieved with the heat conduction equations to gain practical advantages. Our model is also organized effectively to improve the segmentation. The proposed

algorithm has two parts. In the first part, we represent a particular heat conduction problem in the image domain to roughly segment the region of interest. Then, in the second part, geometric heat flow is used to complete the segmentation, by smoothing extracted boundaries and removing noise inside the prior segmented region. Our algorithm is compared with popular active contour models and is tested on synthetic and medical images. Experimental results indicate that our approach works well in noisy conditions. It is also computationally more efficient and easier to control and implement in comparison with active contour models.

## 2 Introduction

There are two main types of shape extraction method that evolve to the target solution: active contours and region growing techniques. We first review these techniques with special consideration of their advantages and practical limitations. We then describe techniques which are based on the use of the heat flow analogy, including the proposed model and its advantages as a segmentation technique.

### 2.1 Related works

#### 2.1.1 Active contours

Active contours (snakes) are curves that evolve to recover object shapes. Active contours can be classified as Parametric Active Contours (PAC) and Geometric Active Contours (GAC) according to their representation. These models mainly differ in their ability to handle multiple object detection, computational efficiency and complexity of implementation.

---

C. Direkoğlu (✉) · M. S. Nixon  
School of Electronics and Computer Science,  
University of Southampton, Southampton SO17 1BJ, UK  
e-mail: cdirekoglu@googlemail.com

A PAC is a parametric curve which is represented explicitly. There are internal and external forces acting on curve. The internal forces control the smoothness and rigidity of the curve with respect to selected parameters. The external force is image dependent and attracts the curve to the object boundary. The first PAC model was introduced by Kass et al. [1]. In this, segmentation is achieved by using gradient vectors of an edge map as an external force. Problems associated with this model are initialization and poor convergence to concave regions. Initialization and convergence to concave region problems were largely solved with the development of a new external force models such as Gradient Vector Flow (GVF) [2] and Vector Field Convolution [3]. However, PAC models can have difficulty with simultaneous detection of multiple objects, since they cannot handle the topological changes such as merging or splitting of the moving curve, because of the explicit representation. To solve this problem, GAC models have been introduced, where the curve is represented implicitly in a level set function (the zero set).

GAC use the level set method [4] for curve evolution. A level set is a real-valued function of the two-dimensional space variables. The contour is represented implicitly in a level set function, which is the zero set. Caselles et al. [5] and Malladi et al. [6] proposed the first GAC model, which uses gradient-based information for segmentation. In this model, the curve evolves in the direction normal with its curvature and balloon force-dependent speed, and stops depending on the gradient information obtained from the image. The curvature-dependent speed has the effect of smoothing the curve, while the balloon force accelerates the motion. The gradient-based GAC can detect multiple objects simultaneously but it has other important problems, which are boundary leakage, noise sensitivity, computational inefficiency and complexity of implementation. Some formulations, [7–9], have been introduced to solve problems with boundary leakage and noise sensitivity by improving gradient-based information. However, they can just increase the tolerance, since gradient-based information is always limited by noise. Several numerical schemes have been proposed to improve the computational efficiency of the level set method, including narrow band [10], fast marching [11] and additive operator splitting [12]. Despite substantial improvements in efficiency, they can be difficult to implement.

Chen and Vese [13] introduced a new GAC model without using edge information. Their model is a particular case of the Mumford-Shah functional [14] and uses regional statistics for segmentation. Their approach especially works well for bimodal images. It is good at handling initialization, noise and boundary leakage but still suffers from computational complexity and difficulty in implementation, because of the level set method. In addition, this

model does not work well for the images with intensity inhomogeneity, which often occurs in real images.

There are different models that have been introduced to segment images with intensity inhomogeneities [15–19]. Li et al. [17, 18] proposed an algorithm based on minimization of region-scalable fitting (RSF) energy to overcome intensity inhomogeneity in images. RSF-based active contour uses the local intensity information (weighted averages of the intensities), which is provided by a Gaussian kernel, for curve evolution and segmentation. However, the local intensity information is sensitive to strong noise and does not provide enough information for accurate segmentation. In addition, RSF-based active contour is implemented with level set method that is difficult and time consuming. Recently, Zhang et al. [19] introduced another energy functional, which is called local image fitting (LIF) energy, to extract the local image information. They proposed LIF-based active contour to overcome intensity inhomogeneity in the images. Their analysis and results show that their method is computationally more efficient than RSF-based active contour, while performing similar segmentation results.

Different types of image forces can be combined with PAC or GAC (level set) models to overcome limitations and improve segmentation. In [20], the Mumford-Shah functional [14] based region force was adapted by PAC to handle noise and initialization problems. Paragios and Deriche [21] unified boundary- and region-based forces and implemented using GAC to improve segmentation and solve initialization and noise problems. Xie and Mirmehdi [22] proposed a magnetic force based on magnetostatic theory using GAC to handle problems with initialization and convergence to concave regions.

Recently, a new class of deformable model that is called Metamorphs [23] has also been proposed. They combine the best features of PAC and GAC and introduce a novel modelling strategies that unify the representation and deformation schemes for shape and intensity. Metamorphs model can also solve the merging problem of PAC.

### 2.1.2 Region growing

Region growing is a procedure that groups pixels or sub-regions into larger regions based on predefined similarity criteria for region growth. The basic approach starts with a seed point and merges neighbouring pixels that have predefined properties similar to the seed, such as intensity [24] or texture [25]. Region growing was also combined with edge information for segmentation [26–28]. Although region growing techniques can detect multiple objects simultaneously and can be more efficient than active contour models, the main problem is selection of the similarity criteria. They also have to use connectivity information to

define the neighbouring pixels in each step of growth. In addition, they may achieve region segmentation with irregular boundaries and holes in the presence of high noise, since they omit smoothing.

## 2.2 Heat flow in image processing and computer vision

The heat flow analogy has been deployed in various ways in image processing and computer vision. It has been used for image smoothing and enhancement; region-based image segmentation; thinning; active contours and motion analysis. Anisotropic diffusion, which was introduced to computer vision by Perona and Malik [29], is the state-of-art image enhancement technique. In [30], the anti-geometric heat flow model was introduced for the segmentation of regions. Here, anti-geometric heat flow is represented as diffusion through the normal direction of edges. In [31], a new thinning algorithm was introduced based on time-reversed isotropic heat flow. A geometric active contour [5, 6] is based on a curve moving in normal direction with its curvature-dependent speed, which is also called level set method [4]. The curve movement in the level set approach is achieved with geometric heat flow [32]. In motion analysis, Makrogiannis and Bourbakis [33] proposed a spatio-temporal anisotropic heat diffusion for motion activity measurement. In [34] and [35], the algorithm combines anisotropic and isotropic heat flow to obtain moving edges.

In this paper, we introduce a novel evolution-based segmentation algorithm based on the heat flow analogy. Our model is new and different because of the evolution technique and organization of segmentation. The evolution is achieved with the heat conduction equations to gain practical advantages. Our model is also organized effectively to improve the segmentation. The proposed algorithm consists of two parts. In the first part, we represent a particular heat conduction problem in the image domain to roughly segment objects of interest. In this problem, we consider a two-dimensional conductive solid body that has uniform conductivity within an isotropic medium. Initial and boundary conditions are given, respectively, by  $T(\mathbf{x}, t = 0) = 0$  and  $T(\mathbf{x}, t) = 0$ , where  $T$  represents the temperature at position  $\mathbf{x} = (x, y)$  and time  $t$ . The given conditions mean that the temperature is initially zero inside the body and the boundary condition is “Dirichlet” that has specified temperature, zero, at the boundary layer for all time. If we initialize a continuous heat source (a positive constant) at any point inside the body, there will be heat diffusion to the other points from the source position as time passes and this will cause temperature increase within the body, except at the boundary layer. This concept is represented in the image domain by using a control function in the heat conduction equation. The control function

is obtained from the region’s statistics of the source location, since we propose to segment the source located region. However, in noisy conditions, we can observe irregular boundaries and holes inside the segmented region. These problems are solved in the second part of the algorithm, which is geometric heat flow. In this part, the segmented image is first converted to binary form and then geometric heat flow is applied to reduce curvature in the boundary, as well as to remove holes inside the segmented region. After a specified number of iterations, the resultant image is thresholded and the final segmentation is obtained. Experimental results indicate that the proposed algorithm works well in noisy conditions without pre-processing. It can detect multiple objects simultaneously, if a heat source is located at the background. It is also computationally more efficient and easier to control and implement in comparison with active contour models. As such, by using physics-based analogies, we can control the segmentation process so as to achieve a result which offers improved segmentation, by a better fit to the image data.

The preliminary version and the summary of this work were presented in [36] and [37], respectively. In this paper, we extend the evaluations and explanations of the proposed algorithm. We have more comprehensive performance and time evaluations for our model and for other models. We also compare our model with one of the latest methods [18] in the active contours context. In addition, we show and discuss how the different regional statistic parameters and the different iteration number for geometric heat flow effect the segmentation in our algorithm.

The rest of the paper is organized as follows: Sect. 2 explains the basic concepts of heat flow. Section 3 represents the proposed heat conduction problem in the image domain. Section 4 discusses the geometric heat flow. Section 5 concerns evaluation and experimental results and finally Sect. 6 is conclusions. A List of acronyms is also given below in Table 1.

**Table 1** List of acronyms

ACWE	Active contours without edges
CF	Control function
GAC	Geometric active contours
GHF	Geometric heat flow
GVF	Gradient vector flow
GVFS	Gradient vector flow snake
PAC	Parametric active contours
RSF	Region-scalable fitting
RSAC	Region-scalable active contour
SSE	Sum of squared error
TF	Temperature front

### 3 Basic concepts of heat flow

Conduction, convection and radiation are three different modes of heat flow. Here, we chose to investigate use of a conduction model, which we found to operate well. Conduction is the flow of heat energy from high- to low-temperature regions due to the presence of a thermal gradient in a body [38]. The change of temperature over time at each point of material is described using the general heat conduction or diffusion equation,

$$\partial T / \partial t = \alpha \Delta T + Q = \alpha (\partial^2 T / \partial x^2 + \partial^2 T / \partial y^2) + Q \quad (1)$$

where  $\partial T / \partial t$  is the rate of change of temperature,  $\Delta$  represents the spatial Laplace operator for the temperature  $T$ ,  $\alpha$  is called thermal diffusivity of the material and a larger values of  $\alpha$  indicate faster heat diffusion through the material.  $Q$  is the source term that applies internal heating. It can be uniformly or non-uniformly distributed over material body. The solution of this equation provides the temperature distribution over the material body, and it depends on time, distance, heat source, properties of material, as well as specified initial and boundary conditions.

Initial conditions specify the temperature distribution in a body, as a function of space coordinates, at the origin of the time coordinate ( $t = 0$ ). Initial conditions are represented as follows:

$$T(\mathbf{x}, t = 0) = F(\mathbf{x}) \quad (2)$$

where,  $\mathbf{x} = (x, y)$  is the space vector for the two-dimensional case and  $F(\mathbf{x})$  is the function that specifies the initial temperature inside the body.

Boundary conditions specify the temperature or the heat flow at the boundaries of the body. There are three general types of boundary conditions: Dirichlet, Neuman and Robin. Here, we explain the Dirichlet and Neuman conditions, which are used in our algorithm. In the Dirichlet condition, temperature is specified along the boundary layer. It can be a function of space and time, or constant. The Dirichlet condition is represented as follows:

$$T(\mathbf{x}, t) = \Phi(\mathbf{x}, t) \quad (3)$$

where  $\Phi(\mathbf{x}, t)$  is the function that specifies the temperature at the boundary layer.

In the Neuman condition, the normal derivative of temperature, heat flow, is prescribed at the boundary surface, and it can be a function of space and time, or constant. The Neuman condition is given in the form.

$$\frac{\partial T(\mathbf{x}, t)}{\partial n} = \Phi(\mathbf{x}, t) \quad (4)$$

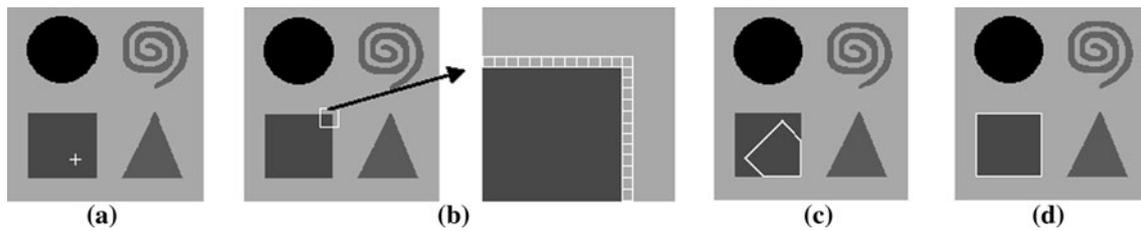
where,  $\Phi(\mathbf{x}, t)$  is the function that specifies the normal derivative of temperature,  $\partial T(\mathbf{x}, t) / \partial n$ , at the boundary

layer. Many heat conduction problems do not have analytical solutions. These problems usually involve geometrical shapes that are mathematically unsuited to representing initial and boundary conditions. However, numerical techniques exist, such as finite differences and finite elements, which are able to handle almost all problems with any complex shapes. The numerical methods yield numerical values for temperatures at selected discrete points within the body and only at discrete time intervals.

The numerical heat conduction problem can be investigated in the image domain, since the image is formed by a set of points, as well as since the image is convenient for the finite difference technique. Each object in image can represent bodies and each pixel in object can represent points within the body.

### 4 Proposed heat conduction problem and representation in image domain

Consider a two-dimensional conductive solid body that has uniform conductivity and is an isotropic medium. Initial and boundary conditions, respectively, are given by  $T(\mathbf{x}, t = 0) = 0$  and  $T(\mathbf{x}, t) = 0$ , where  $T$  represents the temperature at position  $\mathbf{x} = (x, y)$  and time  $t$ . The given conditions mean that the temperature is initially zero inside the body and the boundary condition is Dirichlet that has a specified temperature (zero) at the boundaries. If we initialize a continuous heat source, which is a positive constant, at a point inside the body, there will be heat diffusion to the other points from the source position. As a result of this, all the points inside the body will have temperature values exceeding zero, except the boundary points. This is then an ideal approach for object segmentation in computer images. Let us investigate the proposed problem on a square object that is inside the grey-level image ( $\mathbf{G}$ ), as shown in Fig. 1a. Assume that all the temperature values of the objects and the background are stored in another image, which is represented by  $\mathbf{I}$ , and the initial condition of whole image is zero,  $\mathbf{I}(\mathbf{x}, t = 0) = 0$ . This assumption means that all objects have temperature initially zero inside, as well as at the boundaries. When we initialize a heat source at any pixel inside the square object, as shown in Fig. 1a, there will be heat diffusion to the other pixels from the source position, which will cause temperature to increase. However the temperature at the boundary layer must be kept at zero all the time to obtain the Dirichlet condition, where the boundary layer is defined at the external side of an object as shown in Fig. 1b. To achieve this, we use a control function in the heat conduction equation as given below:



**Fig. 1** Heat conduction modelling in image domain of size  $150 \times 150$ . **a** Source position at  $t = 0$ . **b** Boundary layer illustration. **c** TF at  $t = 30$  (iterations). **d** Final TF at  $t = 69$

$$\frac{\partial \mathbf{I}(\mathbf{x}, t)}{\partial t} = \mathbf{CF}(\mathbf{x}, t)(\alpha \Delta \mathbf{I}(\mathbf{x}, t)) + \mathbf{Q}(\mathbf{x}) \quad (5)$$

where  $\mathbf{I}(\mathbf{x}, t)$  represents an image pixel value in terms of temperature at each point and time,  $\alpha$  is the thermal diffusivity and  $0 \leq \alpha \leq 0.25$  for the numerical scheme to be stable in two-dimensional system [38],  $\Delta$  is Laplace operator for the spatial variables,  $\mathbf{Q}(\mathbf{x})$  is the source term and  $\mathbf{CF}(\mathbf{x}, t)$  is the control function. The control function is obtained from the region statistics of source location on a given grey-level image.

The proposed region statistics model is similar to the one used by Chen and Vese [13]. In this model, the image is divided into two regions, interior and exterior, separated by a contour and the model minimizes the sum of the squares of the differences between the intensity values and the mean of the intensity values inside and outside of the surface of desired object. In our model, the contour is represented by a Temperature Front (TF), where the TF is the boundary of the region that has temperature values exceeding zero. In [13], the active contour model is initialization invariant and attempts to segment whole image, which is a global minimizer. On the other hand, our model attempts to segment the region containing the source, which is a local minimizer.

The control function,  $\mathbf{CF}(\mathbf{x}, t)$ , is formulated as follows:

$$\sigma_{in}(\mathbf{x}, t) = \lambda_{in} |\mathbf{G}(\mathbf{x}) - \mu_{in}|^2 \quad (6)$$

$$\sigma_{out}(\mathbf{x}, t) = \lambda_{out} |\mathbf{G}(\mathbf{x}) - \mu_{out}|^2 \quad (7)$$

where,  $\mathbf{G}(\mathbf{x})$  is the given grey-level image,  $\sigma_{in}(\mathbf{x}, t)$  is the weighted square of the difference between the intensity value and the mean,  $\mu_{in}$ , of the intensity values inside the TF, at each point and time.  $\sigma_{out}(\mathbf{x}, t)$  is the weighted square of the difference between the intensity value and the mean,  $\mu_{out}$ , of the intensity values outside the TF, at each point and time.  $\lambda_{in} > 0$  and  $\lambda_{out} > 0$  are fixed parameters (constants) for regional statistics inside and outside the TF, respectively. Then, the following logical decision is applied at each position and at each time interval

$$\mathbf{CF}(\mathbf{x}, t) = \begin{cases} 1, & \sigma_{in}(\mathbf{x}, t) \leq \sigma_{out}(\mathbf{x}, t) \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

Therefore, the control function allows heat diffusion inside the object of interest and achieves the proposed

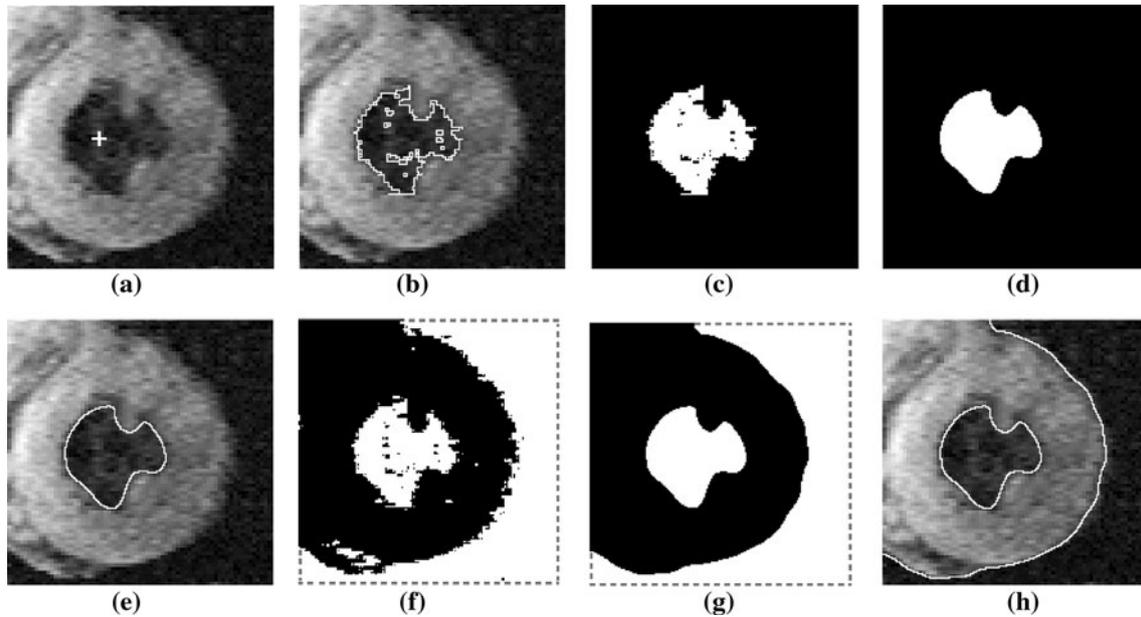
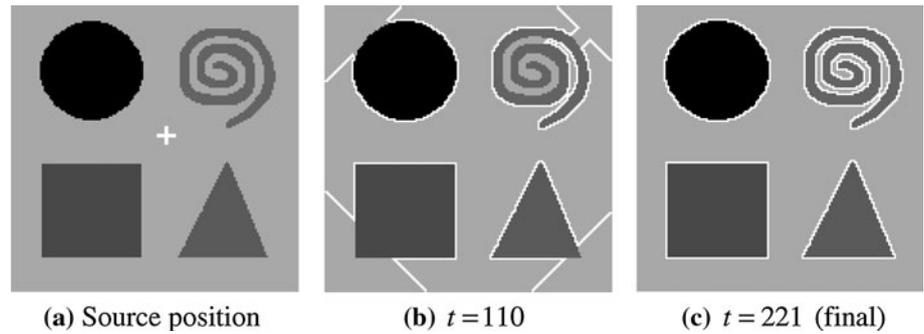
Dirichlet condition on the boundary layer by keeping the temperature value at zero. However, it is better to start this process after a short diffusion time by assuming  $\mathbf{CF}(\mathbf{x}, t) = 1$  at all points. This increases the effective area of initialization and thereby better handling of noisy images. In addition, the heat source must be initialized onto a smooth surface of the object, since the source localization to the edge pixel will give the wrong region statistic for our purpose. Figure 1c and d, respectively, show the evolution and the final position of the TF. However, there is no need to continue diffusion, after the TF reaches its final position. For this reason, the position of the TF is controlled at each specified time interval and when there is no movement, diffusion is terminated. Here, we also note that the regional statistics parameters will be represented with their ratio  $\lambda_{in}/\lambda_{out}$  in our experiments, and in this experiment (Fig. 1)  $\lambda_{in}/\lambda_{out} = 1$ .

One difficulty arises when the region containing the source intersects the image boundary. This problem can be solved by assuming that image is surrounded by a boundary layer, at its periphery, which has temperature value zero for all time (Dirichlet). Figure 2 shows the evolution and the final position of the TF, which has a source location within the background. The result, in Fig. 2, also shows that multiple object detection can be achieved and the heat can diffuse through the narrow regions within the spiral object. As such, heat has been used to detect the background and therefore boundaries of objects.

It is also required to consider the control function when the given image is bimodal. In this case, the control function attempts to segment the whole image while the TF segments the source located region. This is because the control function assigns unity to the pixels that are similar to the inside of the TF and assigns zero to the pixels that are dissimilar.

All the results so far have been on synthetic images without added noise. If we simulate this algorithm on noisy medical images, such as a magnetic resonance image of the left ventricle of a human heart shown in Fig. 3a with the heat source location, we observe some drawbacks in segmentation. The drawbacks are irregular boundaries and holes inside the segmented region, as shown in Fig. 3b, where the  $\lambda_{in}/\lambda_{out} = 1$  in this experiment. These problems

**Fig. 2** TF is moving on background in the image of size  $150 \times 150$ , and  $\lambda_{in}/\lambda_{out} = 1$



**Fig. 3** A magnetic resonance image of the left ventricle of a human heart and illustration of Geometric heat flow (GHF) for the purpose of obtaining smooth boundaries and removing holes inside the prior segmented regions. GHF is applied both to the binary form of the Temperature Front (TF) segmentation,  $\mathbf{B}(x)$ , and to the control function  $\mathbf{CF}(x)$ . The size of the image is  $177 \times 178$ ,  $\lambda_{in}/\lambda_{out} = 1$  and

$t_s = 50$ . **a** Source position. **b** Final position of the TF at  $t = 59$ . **c** Binary form of the segmentation by TF,  $\mathbf{B}(x)$ . **d** Binary form of the final segmentation,  $\mathbf{S}(x)$ , after GHF. **e** Final shape after GHF. **f** Final form of the control function,  $\mathbf{CF}(x)$ . **g** Control function,  $\mathbf{CF}(x)$ , after GHF. **h** Final shape after GHF

are solved by using the heat flow analogy again as described in the next section.

### 5 Geometric heat flow

Geometric heat flow (GHF) is a kind of anisotropic diffusion and is widely used for image denoising and enhancement [32]. It diffuses along the boundaries of image features, but not across them. It derives its name from the fact that, under this flow, the feature boundaries of the image evolve in the normal direction in proportion to their curvature. Thus GHF decreases the curvature of shapes while removing noise. GHF equation is obtained with the following considerations:

Edge directions are related to the tangents of the feature boundaries of an image  $\mathbf{B}$ . Let  $\eta$  denote the direction normal to the feature boundary through a given point (the gradient direction), and let  $\tau$  denote the tangent direction. These directions can be written in terms of the first derivatives of the image,  $\mathbf{B}_x$  and  $\mathbf{B}_y$ , as

$$\eta = \frac{(\mathbf{B}_x, \mathbf{B}_y)}{\sqrt{\mathbf{B}_x^2 + \mathbf{B}_y^2}}, \quad \tau = \frac{(-\mathbf{B}_y, \mathbf{B}_x)}{\sqrt{\mathbf{B}_x^2 + \mathbf{B}_y^2}}, \quad (9)$$

Since  $\eta$  and  $\tau$  constitute orthogonal directions, the rotationally invariant Laplacian operator can be expressed as the sum of the second-order spatial derivatives,  $\mathbf{B}_{\eta\eta}$  and  $\mathbf{B}_{\tau\tau}$ , in these directions and the linear heat conduction equation can be written without using the source term,

$$\frac{\partial \mathbf{B}}{\partial t} = \alpha \Delta \mathbf{B} = \alpha (\mathbf{B}_{\eta\eta} + \mathbf{B}_{\tau\tau}) \tag{10}$$

Omitting the normal diffusion, while keeping the tangential diffusion yields the GHF equation as

$$\frac{\partial \mathbf{B}}{\partial t} = \alpha \mathbf{B}_{\tau\tau} = \alpha \frac{(\mathbf{B}_{xx}\mathbf{B}_y^2 - 2\mathbf{B}_{xy}\mathbf{B}_x\mathbf{B}_y + \mathbf{B}_{yy}\mathbf{B}_x^2)}{(\mathbf{B}_x^2 + \mathbf{B}_y^2)}. \tag{11}$$

The equation above can also be written as follows to show that feature boundaries of the image evolve in the direction normal in proportion to their curvature  $\kappa$ :

$$\begin{aligned} \frac{\partial \mathbf{B}}{\partial t} = \alpha \mathbf{B}_{\tau\tau} &= \alpha \frac{(\mathbf{B}_{xx}\mathbf{B}_y^2 - 2\mathbf{B}_{xy}\mathbf{B}_x\mathbf{B}_y + \mathbf{B}_{yy}\mathbf{B}_x^2)}{(\mathbf{B}_x^2 + \mathbf{B}_y^2)^{3/2}} \\ &\times \sqrt{(\mathbf{B}_x^2 + \mathbf{B}_y^2)} \\ &= \kappa |\nabla \mathbf{B}|. \end{aligned} \tag{12}$$

In our model, GHF is used to decrease the curvature for the purpose of obtaining smooth boundaries and removing holes that appear because of noise. This is achieved as follows. First, a segmented region is converted to a binary form as given below and also shown in Fig. 3c,

$$\mathbf{F}(\mathbf{x}) = \begin{cases} 1, & \mathbf{I}(\mathbf{x}) > 0 \\ 0, & \mathbf{I}(\mathbf{x}) = 0 \end{cases} \tag{13}$$

where  $\mathbf{I}(\mathbf{x})$  is the temperature distribution after terminating diffusion and  $\mathbf{F}(\mathbf{x})$  is the binary form of the segmented image that assigns unity to the region of interest. This binary form represents initial condition for GHF,  $\mathbf{B}(\mathbf{x}, t = 0) = \mathbf{F}(\mathbf{x})$ . The boundaries of the image are insulated with homogeneous Neuman condition,  $d\mathbf{B}/dn = 0$ , which means there is no heat flow in, or out, from the boundaries. The GHF problem is defined below,

$$\frac{\partial \mathbf{B}}{\partial t} = \alpha \frac{(\mathbf{B}_{xx}\mathbf{B}_y^2 - 2\mathbf{B}_{xy}\mathbf{B}_x\mathbf{B}_y + \mathbf{B}_{yy}\mathbf{B}_x^2)}{(\mathbf{B}_x^2 + \mathbf{B}_y^2)} \tag{14}$$

$$\mathbf{B}(\mathbf{x}, t = 0) = \mathbf{F}(\mathbf{x}), \quad \text{initial condition}$$

$$\frac{\partial \mathbf{B}(\mathbf{x}, t)}{\partial n} = 0, \quad \text{boundary condition}$$

GHF is applied to the  $\mathbf{B}(\mathbf{x})$  until the specified time (number of iterations) and finally the resulting image is thresholded to obtain the final segmentation. The process is formulated below:

$$\mathbf{S}(\mathbf{x}) = \begin{cases} 1, & GHF(\mathbf{B}(\mathbf{x}), t_s) \geq 0.5 \\ 0, & GHF(\mathbf{B}(\mathbf{x}), t_s) < 0.5 \end{cases} \tag{15}$$

where,  $t_s$  is the number of iterations and  $\mathbf{S}(\mathbf{x})$  is the binary form of the final segmentation, which assigns unity to the

region of interest. The final segmentation is shown in Fig. 3d and e, where  $t_s = 50$  for this illustration. The selection of  $t_s$  depends on the user and it is determined due to the noise level of the image. However, as  $t_s$  increases, extracted shape evolves to a circle, then to a point and then it is lost. For this reason, we should avoid using large values for  $t_s$ .

Since the illustrated human heart image seems bimodal, we can also consider the final form of the control function as shown in Fig. 3f. To smooth boundaries and remove holes, we simply continue with Equation 14 and observe the result in Fig. 3g and h.

### 6 Evaluation and experimental results

In evaluation, our model is denoted by TF + GHF, since the TF roughly segments the region of interest and then the GHF is used to complete the segmentation. Similarly, the segmentation by the control function (CF) followed by the GHF is denoted by CF + GHF. In this section, we first present the evaluation of TF + GHF on segmentation and then the evaluation for the computational efficiency of our algorithm, finishing with some illustrative examples on medical images. Note that in this section we are using many acronyms, so for convenience a list of abbreviations is given at the beginning of this paper.

Segmentation by TF + GHF is compared with the active contour without edges (ACWE) [13], region-scalable fitting active contour (RSAC) [18] and gradient vector flow snake (GVFS) [2]. The evaluation is done on a star-shaped object with varying normal distributed noise  $N_d(\mu, \sigma^2)$ , as shown at the top row in Fig. 5. The sum of squared error (SSE) is employed to quantify the performance of each algorithm.

$$SSE = \sum_{x=1}^M \sum_{y=1}^N (\mathbf{S}_{x,y} - \mathbf{O}_{x,y})^2 \tag{16}$$

where,  $\mathbf{S}$  is the binary segmented image and  $\mathbf{O}$  is the actual (ground truth) binary segmented image of size  $M \times N$ . The quantity of noise is considered in terms of standard deviation  $\sigma$  with zero mean.

ACWE is a region-based GAC model that is implemented by a level set function [13]. It applies global minimization to segment bimodal images as a whole. However, in this evaluation, we choose the biggest segmented region, since we are concerned with the star-shaped object segmentation. Otherwise, it will cause very high errors in noisy conditions because of the noise outside the star object. To evaluate ACWE, we use a Matlab implementation given in [39], and we note that this is a non-optimal Matlab framework. In this evaluation, the selected

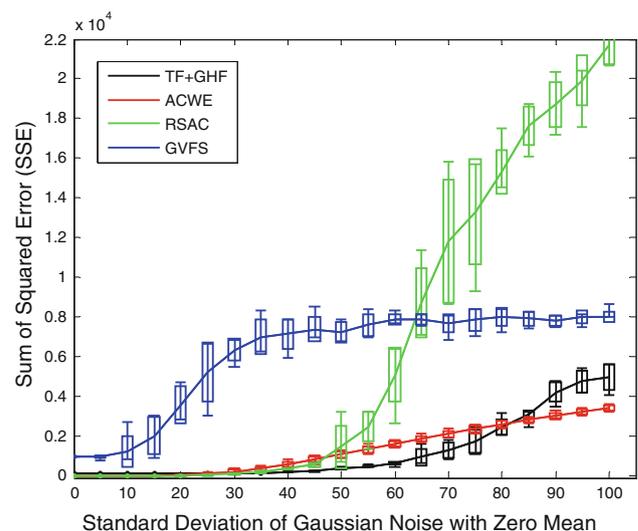
parameter values for ACWE are  $\lambda_{in} = \lambda_{out} = 1$  (parameters for regional statistics),  $v_A = 0$  (the area parameter),  $h_A = 1$  (the step space),  $\Delta t = 0.1$  (the time space),  $\varepsilon = 1$  (the parameter for the Heaviside and Dirac delta functions) and  $\mu_A = 0.01 \times 255^2$  (the length parameter).

RSAC is a region-based GAC model which is implemented with level set method [18]. The proposed region-based model uses intensity information in local regions as a controllable scale. A Gaussian kernel is used to compute local intensity information. In this evaluation, we choose the biggest segmented region, since we are concerned with the star-shaped object segmentation. Otherwise, it will cause very high errors in noisy conditions because of the noise outside the star object. To evaluate RSAC, we use a Matlab implementation given in [40], and we note that this is a non-optimal Matlab framework. In this evaluation, the selected parameter values for RSAC are  $\lambda_{in} = \lambda_{out} = 1$  (parameters for regional statistics),  $\sigma = 8$  (scale parameter in Gaussian kernel),  $r = 1$  (coefficient of the level set regularization term),  $\Delta t = 0.1$  (the time space),  $\varepsilon = 1$  (the parameter in the definition of smoothed Dirac function) and  $\mu_R = 0.01 \times 255^2$  (the length parameter).

GVFS is a gradient-based PAC model that uses GVF as an external force [2]. To evaluate GVFS, we use Matlab implementation given in [41], and this is also a non-optimal Matlab framework. In this evaluation, the selected parameter values for GVFS are  $\alpha_G = 0.25$  (smoothness parameter of the contour),  $\beta_G = 0$  (rigidity parameter of the contour),  $\gamma_G = 0.6$  (external energy parameter of the contour), and  $\mu_G = 0.2$  (in diffusion of gradient vectors),  $\Delta t = 1$  (the time interval). In addition, we use 80 iterations to diffuse gradient vectors.

In our algorithm, we use an explicit scheme of finite differences in the first and the second parts. We use Matlab for the implementation as well. In this evaluation, the selected parameter values for TF + GHF are  $\alpha = 0.25$  (thermal diffusivity),  $\lambda_{in}/\lambda_{out} = 1$  (ratio of the parameters for regional statistics),  $Q = 5$  (the energy generated from the source position per unit time interval),  $\Delta t = 1$  (the time interval),  $\Delta x = \Delta y = 1$  (the spatial intervals),  $t_s = 10$  (specified time for GHF). In addition, we start to use regional statistics after  $t = 10$  to increase the number of samples inside the TF and in each ten iterations we control the movement of TF to determine the termination of the first part.

In the evaluations for TF + GHF, the contours and the heat source are initialized inside the star objects. Figure 4 shows the performance of TF + GHF, ACWE, RSAC and GVFS for the segmentation of the star object. The graphs are obtained over five applications of each algorithm, where the rectangle represents standard deviation from the mean value and error bar represents minimum and maximum values at each data point. In each application, the



**Fig. 4** Performance of TF + GHF, ACWE, RSAC and GVFS on the star-shaped object. The graphs show the rectangle representing standard deviation from the mean value and error bar representing minimum and maximum values of the SSE

obtained graphs are smoothed by applying local averaging to the data points in six nearest neighbourhoods. The star object has branches, which narrow until they are one pixel wide. Half of the star object has different intensity, to create weak edges inside the object.

It is observed that TF + GHF and ACWE perform much better than GVFS at all noise levels. The reason for this is that TF + GHF and ACWE use region-based algorithms; on the other hand GVFS uses a gradient-based algorithm, which is very sensitive to the noisy conditions. RSAC performs better than GVFS until  $\sigma \cong 65$  since RSAC is also using regional information (statistics of local regions). After  $\sigma \cong 65$ , RSAC performs worse than GVFS, because RSAC relies on the information obtained from the statistics of local regions that is sensitive to higher noise levels. RSAC performs very poorly at higher noise levels. TF + GHF, ACWE and RSAC can handle topological changes, which GVFS cannot. It is also observed that GVFS cannot flow into the narrow regions of the star object. TF + GHF, ACWE and RSAC can flow into the narrow regions. RSAC performs similar to ACWE until  $\sigma \cong 30$  and then performs better than ACWE between  $\sigma \cong 30$  and  $\sigma \cong 48$ . After  $\sigma \cong 48$ , RSAC segments worse than ACWE since RSAC is sensitive to higher noise levels. RSAC and ACWE perform slightly better than TF + GHF until  $\sigma \cong 30$ . This appears to be due to the smoothing operation in TF + GHF. GHF attempts to smooth the original shape and cause errors in TF + GHF, when there is no noise or low noise in the image, since  $t_s$  is fixed in the evaluation. After  $\sigma \cong 30$ , TF + GHF has better performance than RSAC since TF + GHF uses information extracted from whole regions inside and outside the TF. On

the other hand, RSAC uses information extracted from local regions inside and outside the contour. From  $\sigma \cong 30$  to  $\sigma \cong 80$ , TF + GHF segments better than ACWE. The main reason is the smoothing operation in TF + GHF. TF + GHF applies smoothing after rough segmentation without any relation to the regional statistic constraints, while ACWE uses smoothness constraint with regional statistic constraints during the segmentation. After  $\sigma \cong 80$ , it is seen that ACWE shows better performance than TF + GHF, because ACWE segments many regions outside the star region in the presence of high noise and then some of the segmented noise remains connected to the original region when we select the biggest region.

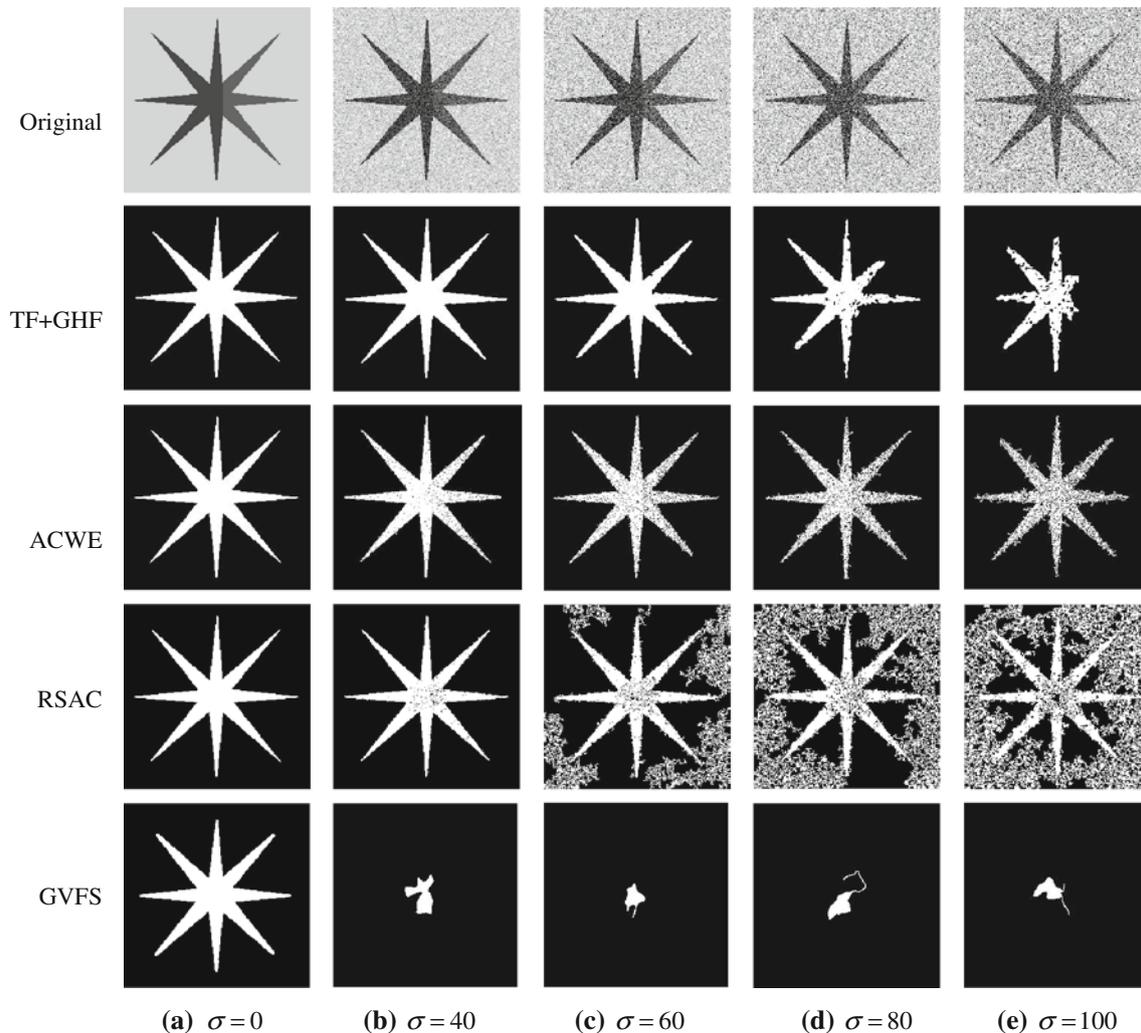
The SSE of RSAC has significant variations after  $\sigma \cong 50$ , and the variations are high in comparison with the other algorithms. The SSE of GVFS starts to vary after  $\sigma \cong 5$ , and the variations are high between  $\sigma \cong 10$  and  $\sigma \cong 45$  in comparison with the other algorithms in this interval. For ACWE, we observe variation in the SSE after  $\sigma \cong 25$ , and the variations look similar at all the remaining noise levels. The variations of SSE of ACWE are less than other algorithms at high noise levels. For TF + GHF, the SSE varies slightly between  $\sigma \cong 40$  and  $\sigma \cong 60$ . After  $\sigma \cong 65$ , we can observe significant variations in the SSE. Figure 5 shows some of the results for TF + GHF (second row), ACWE (third row), RSAC (fourth row) and GVFS (fifth row).

Results also show the effectiveness and the computational efficiency of our algorithm in comparison with the GVFS and ACWE. The following evaluation results are obtained by using MATLAB 7.0 on a Pentium IV computer, which runs Windows XP operating system with 3.2 GHz CPU and 3 GB RAM.

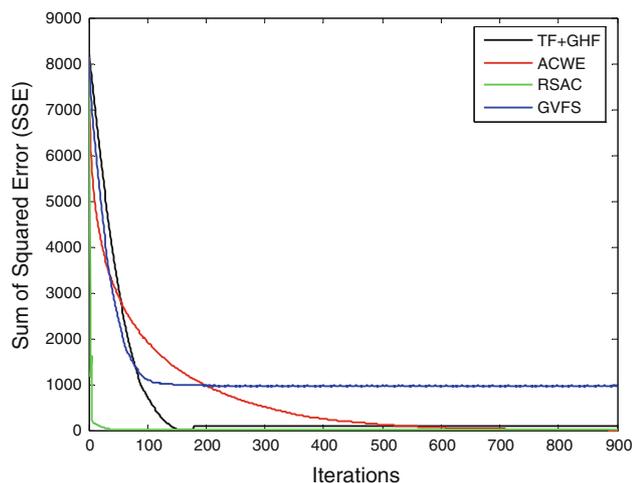
First we investigate how the SSE of TF + GHF, of ACWE, of RSAC and of GVFS changes, on the original (no noise) and on the noisy star object, as iteration number increases (as the contour evolves). We initialize contours at the centre of the star object and arrange their sizes to have SSE  $\cong 8,000$  at the beginning. Figure 6 shows SSE of TF + GHF, of ACWE, of RSAC and of GVFS with respect to increasing iteration numbers on the original star object, shown in Fig. 5a. Table 2 also shows the number of iterations required to stabilize SSE, remaining SSE after stabilization and CPU time required for stabilization for each algorithm on the original star object. The final segmentation by each algorithm can be seen in Fig. 5a as well. We observe that GVFS stabilizes at iteration  $\cong 170$  with SSE  $\cong 1,000$  and CPU time = 57.63 s. For further iterations GVFS remains at that SSE, because it cannot move through the narrow regions and cannot achieve complete segmentation. RSAC stabilizes at iteration  $\cong 121$  with SSE = 2 and CPU time = 1.33 min. ACWE achieves complete segmentation at iteration  $\cong 890$  with SSE = 0 and CPU

time = 5.92 min. Finally, TF + GHF achieves segmentation at iteration  $\cong 180$  with SSE  $\cong 93$  and CPU time = 7.17 s. When the number of iteration is about 180, the SSE of TF + GHF suddenly increases and remains some error. The reason is the smoothing operation in TF + GHF. GHF is applied, at the end of first part in our algorithm, with fixed iteration number (10 iterations) and causes errors when there is no noise on the object. It is also observed that TF + GHF achieves segmentation with the lowest CPU time. Although GVFS segments a smaller region, it requires CPU time more than TF + GHF. It must be noted that GVFS also needs 80 iterations to diffuse gradient vectors of the image, which is the image-dependent force it uses, before contour evolution. There is an also big difference in CPU times of ACWE and TF + GHF. ACWE is computationally expensive since it uses the level set method for curve evolution. TF + GHF also performs segmentation faster than RSAC, since it is also implemented with level set method and since the computation of the RSF energy is time consuming in this model. RSAC performs faster than ACWE, one of the reasons RSAC avoids expensive re-initialization (regularization) process during the evolution. In RSAC, the regularity of level set function is intrinsically preserved by the level set regularization term.

Figure 7 shows SSE of TF + GHF, of ACWE, of RSAC and of GVFS with respect to increasing iteration numbers on the noisy star object, shown in Fig. 5b, which is corrupted by Gaussian noise of  $\sigma = 40$ . Table 3 also shows the number of iterations required to stabilize, remaining SSE after stabilization and CPU time required for each algorithm on the noisy star object. The final segmentation by each algorithm can be seen in Fig. 5b as well. It is observed that as the iteration number increases, GVFS stabilizes at iteration  $\cong 18$  with SSE  $\cong 6,950$  and CPU time 10.57 s. It achieves poor segmentation because of the limitation of the gradient-based information to the noise. RSAC completes segmentation at iteration  $\cong 454$  with SSE  $\cong 255$  and CPU time 5.32 min. ACWE achieves segmentation at iteration  $\cong 1,100$  with SSE  $\cong 570$  and CPU time = 7.29 min. Finally, TF + GHF achieves segmentation at iteration  $\cong 180$  with SSE  $\cong 188$  and CPU time = 7.26 s. In TF + GHF, we observe sudden decrease in SSE before stabilizing. This is because we apply a fixed number of iterations of GHF (10 iterations) to remove noise and smooth shape at the end of our algorithm. It is observed that TF + GHF achieves segmentation with minimum SSE and again with the lowest CPU time. Although GVFS has poor performance because of noise and segments small region, TF + GHF has slightly smaller CPU time than GVFS. TF + GHF performs much faster than ACWE and RSAC. TF + GHF also has the



**Fig. 5** Results for TF + GHF (second row), ACWE (third row), RSAC (fourth row) and GVFS (fifth row) with respect to increasing Gaussian noise in the image of size  $256 \times 256$



**Fig. 6** SSE for TF + GHF, ACWE, RSAC and GVFS with respect to increasing iteration numbers from the initial positions to the final segmentations on the original star object

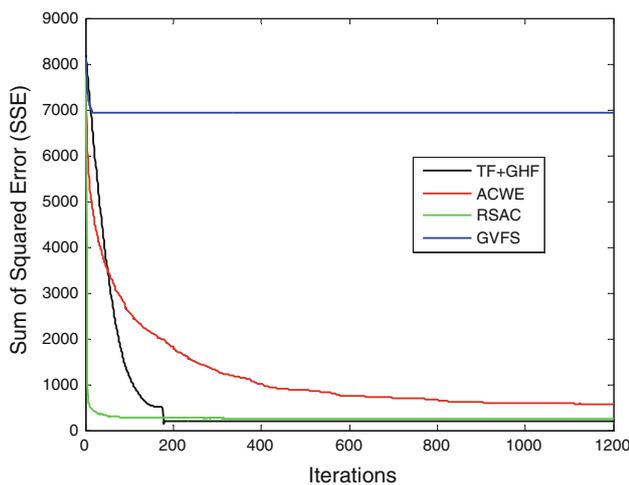
lowest SSE in this evaluation, which means TF + GHF performs better than other algorithms in this segmentation.

It is important to note that rest of the experiments, in this paper, are conducted by same computer and using Matlab 7.0, as described before, except the RAM is 1 GB.

Figure 8 shows the segmentation of pulmonary arterial branches in the chest image by TF + GHF, GVFS, RSAC and ACWE. The initial contour for GVFS, RSAC and ACWE, and the source position for TF + GHF are shown in Fig. 8a. Figure 8b shows the segmentation by TF + GHF in the given image with black contour; however, the segmented arterial branches are not visible with this illustration and the segmentation is also shown with the binary form in Fig. 8c. Figure 8d and e show segmentation by GVFS, respectively, with the result superimposed as the black contour in the image and the binary form. Figure 8f, g shows segmentation by ACWE, respectively, with black contour superimposed and the binary form. Figure 8h, i

**Table 2** Number of iterations required to stabilize, remaining SSE after stabilization and CPU time required to stabilize for TF + GHF, ACWE, RSAC and GVFS on the original star object

Original star object (no noise)			
Algorithm	Num. of iteration to stabilize	Remaining SSE	CPU Time
TF + GHF	180	93	7.17 s
ACWE	890	0	5.92 min
RSAC	121	2	1.33 min
GVFS	170	1,000	57.63 s



**Fig. 7** SSE for TF + GHF, ACWE, RSAC and GVFS with respect to increasing iteration numbers from the initial positions to the final segmentations on the noisy star object

**Table 3** Number of iterations required to stabilize, remaining SSE after stabilization and CPU time required to stabilize for TF + GHF, ACWE, RSAC and GVFS on the noisy star object

Noisy star object ( $\sigma = 40$ )			
Algorithm	Num. of iteration to stabilize	Remaining SSE	CPU Time
TF + GHF	180	188	7.26 s
ACWE	1,100	570	7.29 min
RSAC	454	255	5.32 min
GVFS	18	6,950	10.57 s

shows segmentation by RSAC with black contour superimposed and in the binary form, respectively. It is observed that TF + GHF segments the desired arterial branches better than GVFS. By this result, TF + GHF can easily handle topological changes and flow into the arterial branches with CPU = 7.85 s. However, GVFS cannot handle topological changes and cannot flow into the arterial

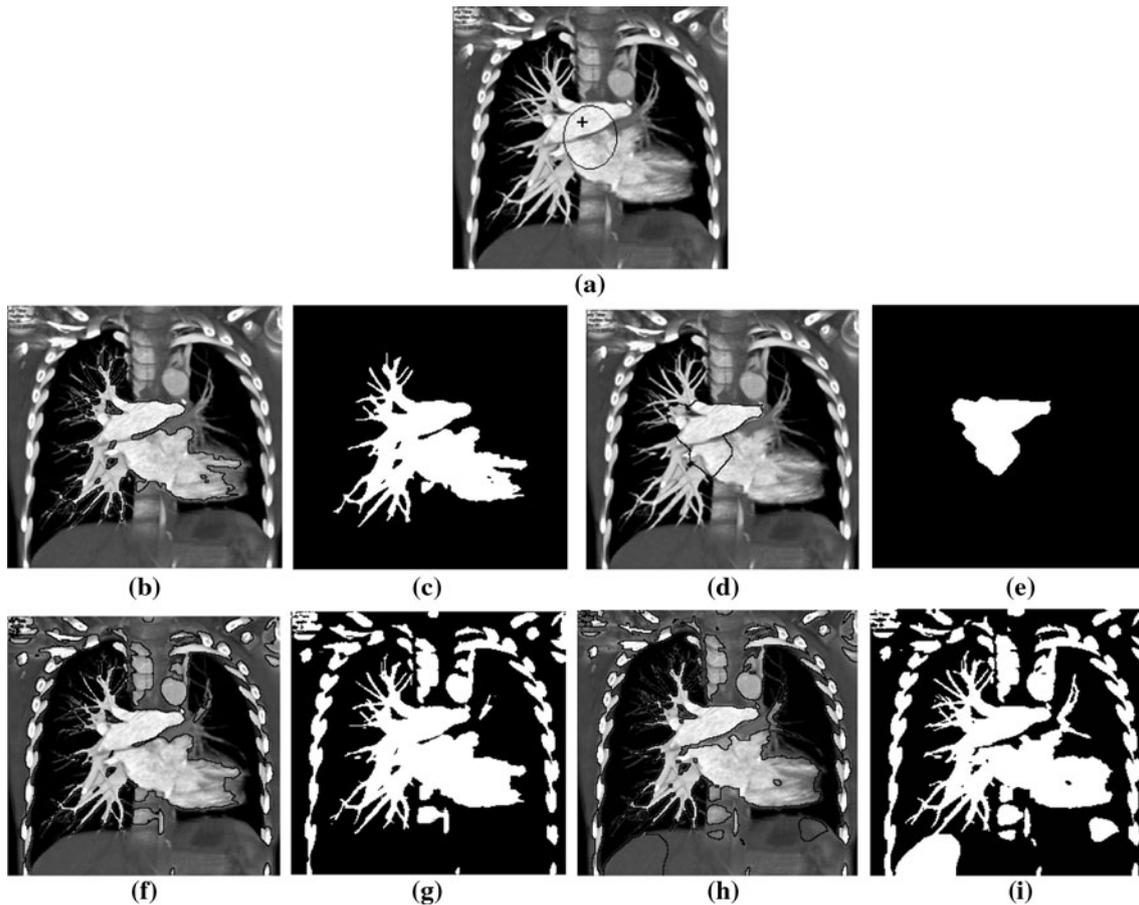
branches. Although GVFS segments a smaller region than TF + GHF, the CPU = 9.23 s, which is more than for TF + GHF. If we look at segmentation by ACWE, we see that ACWE attempts to segment whole image. Here we focus on segmentation of desired region, which is pulmonary arterial branches, and compare with the TF + GHF. TF + GHF and ACWE have similar segmentation results on the desired region. However, ACWE achieves this segmentation in 24.75 min, while TF + GHF achieves in 7.85 s. This big difference in CPU time appears because of the computational complexity of ACWE that is implemented with level sets. RSAC also attempts to segment whole image instead of desired region. If we focus on segmentation of the desired region, we can observe inaccurate segmentation. RSAC segments a darker region that remains connected to the desired region. RSAC achieves this segmentation in 7.09 min, while TF + GHF achieves in 7.85 s. The CPU times of the algorithms for the pulmonary arterial branches are also shown in Table 4.

Figure 9 shows the segmentation of a bimodal human lung image by TF + GHF, CF + GHF and ACWE, where the initial contour for ACWE and the source position for TF + GHF are shown in Fig. 9a. Figure 9b and c, respectively, show the segmentation by TF + GHF and CF + GHF with white contour in the image. Figure 9d shows the segmentation by ACWE and Fig. 9e shows the segmentation by RSAC. It is observed that TF + GHF and CF + GHF achieve segmentation with CPU = 1.96 s, ACWE achieves with CPU = 15.92 min and RSAC achieves with 11.85 s. The CPU times of our algorithm, ACWE and RSAC for the human lung image are also shown in Table 5. It is also observed that CF + GHF can extract feature boundaries better than ACWE and RSAC.

We also discuss how the different regional statistic parameters ( $\lambda_{in}/\lambda_{out}$ ) and the different iteration number for GHF ( $t_s$ ) effect the segmentation.

Figure 10 shows the segmentation of the human lung image by TF + GHF and CF + GHF with respect to the increasing  $\lambda_{in}/\lambda_{out}$ , while keeping  $t_s = 15$ . The heat source is located at the same position shown in Fig. 9(a) and  $\lambda_{in}/\lambda_{out} = 1$  in that experiment. In this experiment, the selected values for  $\lambda_{in}/\lambda_{out} = 0.5, 2, 4, 6$  and the segmentation results by TF + GHF and CF + GHF are given in Fig. 10a–d, respectively. It is observed that as  $\lambda_{in}/\lambda_{out}$  increases from 0.5 to 6, the selectivity for segmentation increases.

Finally, Fig. 11 shows the segmentation of the human lung image by TF + GHF and CF + GHF with respect to the increasing  $t_s$ , while keeping  $\lambda_{in}/\lambda_{out} = 1$ . The heat source is located at the same position shown in Fig. 9a and  $t_s = 15$  in that experiment. In this experiment, the selected values for  $t_s = 5, 10, 20, 25$  and the segmentation results



**Fig. 8** Segmentation of pulmonary arterial branches in the chest image of size  $259 \times 250$  by TF + GHF, GVFS, ACWE and RSAC. **a** Initial contour and the source position. **b** Segmentation by TF + GHF is shown by the black contour on the image. All the parameters are same as in evaluation except  $t_s = 5$ . (CPU = 7.85 s). **c** Segmentation by TF + GHF is in binary form. **d** Segmentation by GVFS is shown with black contour on the image. All the parameters are same as in evaluation except the iteration to diffuse gradient

vectors is 70. (CPU = 9.23 s). **e** Segmentation by GVFS is in binary form. **f** Segmentation by ACWE is shown with black contour on the image. All the parameters are same as in evaluation except the length parameter  $\mu = 0.08 \times 255^2$ . (CPU = 24.75 min). **g** Segmentation by ACWE is in binary form. **h** Segmentation by RSAC is shown with black contour on the image. (CPU = 7.09 min). **i** Segmentation by RSAC is in binary form

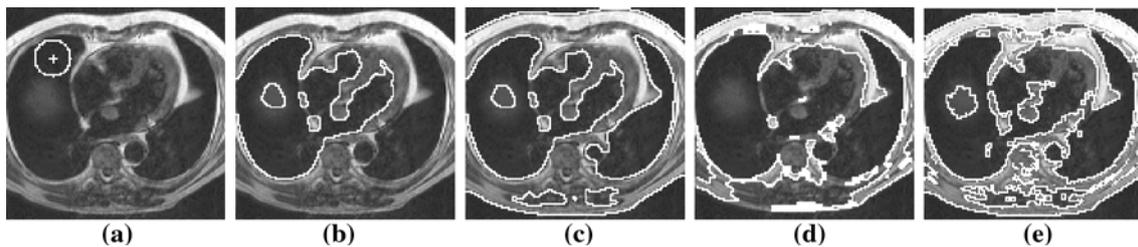
**Table 4** The CPU times of the algorithms for the pulmonary arterial branches in the chest image

Pulmonary Image	
Algorithm	CPU time
TF + GHF	7.85 s
GVFS	9.23 s
ACWE	24.75 min
RSAC	7.09 min

by TF + GHF and CF + GHF are shown in Fig. 11a–d, respectively. It is observed that as  $t_s$  increases, we obtain smoother segmentation, as well as remove noise and smaller regions.

## 7 Conclusions

We have presented a novel segmentation algorithm based on using the heat flow analogy. In the first part of the algorithm, we roughly extract the desired feature boundaries by representing a chosen heat conduction analysis in the image domain. The representation in image domain is achieved by using a control function (CF) in the heat conduction equation. This formulation also provides advantage when the given image is bimodal, since CF attempts to segment whole image in this case. In the second part, we use geometric heat flow (GHF) to tune the curvature of the extracted feature boundaries and remove possible noise that arises from the first part of the segmentation. Evaluation results indicate that Temperature Front (TF) + GHF has better performance than gradient



**Fig. 9** Segmentation of human lung image of size  $123 \times 118$  by TF + GHF, CF + GHF and ACWE. **a** Initial contour and the source position. **b** Segmentation by TF + GHF is shown with white contour on the image. All the parameters are same as in evaluation except  $t_s = 15$  (CPU = 1.96 s). **c** Segmentation by CF + GHF.  $t_s = 15$ .

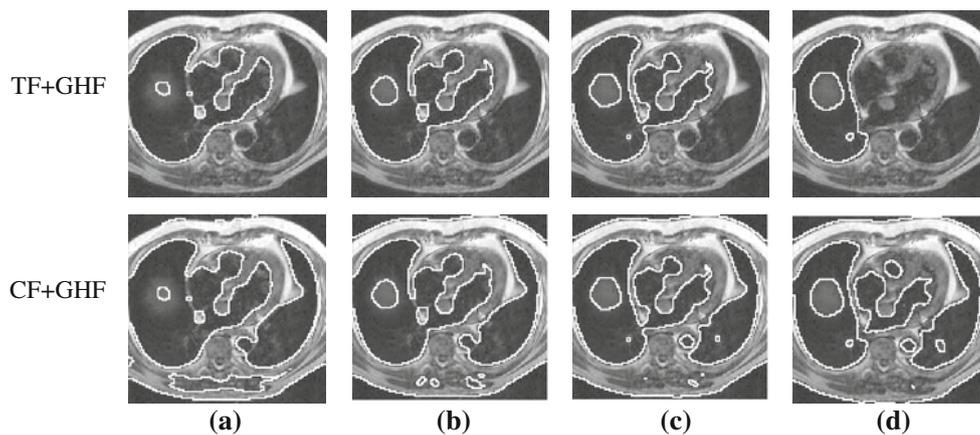
(CPU = 1.96 s). **d** Segmentation by ACWE. All the parameters are same as in evaluation except the length parameter  $\mu = 0.08 \times 255^2$ . (CPU = 15.92 min). **e** Segmentation by RSAC. All the parameters are same as in evaluation except the length parameter  $\mu = 0.001 \times 255^2$  and sigma = 9. (CPU = 11.85 s)

**Table 5** The CPU times of the algorithms for the human lung image

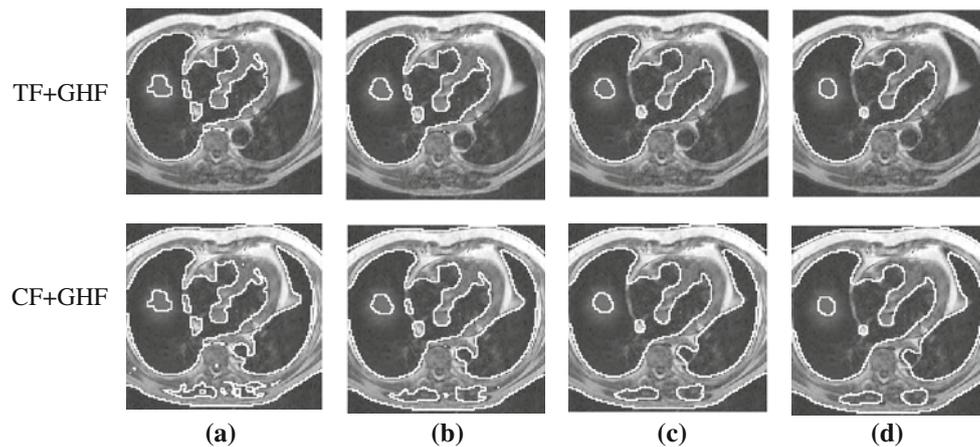
Human lung image	
Algorithm	CPU time
TF + GHF and CF + GHF	1.96 s
ACWE	15.92 min
RSAC	11.85 s

vector flow snake (GVFS), region-scalable fitting active contour (RSAC) and active contour without edges (ACWE) with respect to increasing Gaussian noise. For the bimodal images, CF + GHF has better performance than ACWE and RSAC. These improvements are achieved by effective organization of our algorithm; TF + GHF segments better than GVFS because of using region-based information instead of gradient-based, which is sensitive to noise.

TF + GHF and CF + GHF segment better than ACWE because of the smoothing operation of shape. TF + GHF and CF + GHF apply smoothing after rough segmentation without any relation to the regional statistic constraints, while ACWE uses smoothness constraint with regional statistic constraints during the segmentation. TF + GHF and CF + GHF segment better than RSAC, since the TF + GHF uses intensity information in whole regions (inside and outside the TF) while RSAC uses intensity information in local regions (inside and outside the contour). In addition, TF + GHF and CF + GHF are computationally more efficient and effective than GVFS, RSAC and ACWE based on the simulation results. Especially, there are significant differences between our model and level set based models (ACWE and RSAC) in computational efficiency. The main reason is the complexity of the level set method. As such, the heat analogy can be deployed with success for shape extraction in images.



**Fig. 10** Segmentation of human lung image by TF + GHF and CF + GHF with respect to the increasing  $\lambda_{in}/\lambda_{out}$  (ratio of regional statistics parameters). **a**  $\lambda_{in}/\lambda_{out} = 0.5$ , **b**  $\lambda_{in}/\lambda_{out} = 2$ , **c**  $\lambda_{in}/\lambda_{out} = 4$ , **d**  $\lambda_{in}/\lambda_{out} = 6$



**Fig. 11** Human lung image and segmentation by TF + GHF and CF + GHF with respect to the increasing  $t_s$  (iteration number of GHF). **a**  $t_s = 5$ , **b**  $t_s = 10$ , **c**  $t_s = 20$ , **d**  $t_s = 25$

## References

- Kass M, Witkin A, Terzopoulos D (1987) Snakes: active contour models. *Int J Comput Vision* 321–331
- Xu C, Prince JL (1998) Snakes, shapes and gradient vector flow. *IEEE Trans Image Process* 7(3):359–369
- Li B, Acton ST (2007) Active contour external force using vector field convolution for image segmentation. *IEEE Trans Image Process* 16(8):2096–2106
- Osher S, Sethian JA (1988) Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations. *J Comput Phys* 79(1):12–49
- Caselles V, Catte F, Coll T, Dibos F (1993) A geometric model for active contours. *Numerische Mathematik* 66:1–31
- Malladi R, Sethian JA, Vemuri BC (1995) Shape modeling with front propagation: a level set approach. *IEEE Trans Pattern Anal Mach Intell* 17(2):158–175
- Caselles V, Kimmel R, Sapiro G (1997) Geodesic active contours. *Int J Comput Vision* 22(1):61–79
- Siddiqi K, Lauziere Y, Tannenbaum A, Zucker S (1998) Area and length minimizing flows for shape segmentation. *IEEE Trans Image Process* 7(3):433–443
- Xie X, Mirmehdi M (2004) RAGS: region-aided geometric snake. *IEEE Trans Image Process* 13(5):640–652
- Adalsteinsson D, Sethian J (1995) A fast level set method for propagating interfaces. *J Comput Phys* 118(2):269–277
- Sethian J (1999) *Level set methods and fast marching methods*. Cambridge Univ Press, New York
- Weickert J, Romeny BMTH, Viergever MA (1998) Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Trans Image Process* 7(3):398–410
- Chan T, Vese L (2001) Active Contours without Edges. *IEEE Trans Image Process* 10(2):266–277
- Mumford D, Shah J (1989) Optimal approximation by piecewise smooth functions and associated variational problems. *Commun Pure Appl Math* 42(4):577–685
- Vese L, Chan T (2002) A multiphase level set framework for image segmentation using the Mumford and Shah Model. *Int J Comput Vision* 50:271–293
- Tsai A, Yezzi A, Willsky AS (2001) Curve evolution implementation of the mumford-shah functional for image segmentation, denoising, interpolation, and magnification. *IEEE Trans Image Process* 10(8):1169–1186
- Li C, Kao C, Gore J, Ding Z (2007) Implicit active contours driven by local binary fitting energy. *IEEE IntConf Comput Vision Pattern Recognit* 1–7
- Li C, Kao C, Gore J, Ding Z (2008) Minimization of region-scalable fitting energy for image segmentation. *IEEE Trans Image Process* 17(10):1940–1949
- Zhang K, Song H, Zhang L (2010) Active contours driven by local image fitting energy. *Pattern Recognit* 43:1199–1206
- Zimmer C, Olivo-Marin JC (2005) Coupled parametric active contours. *IEEE Trans Pattern Anal Mach Intell* 27(11):1838–1842
- Paragios N, Deriche R (2000) Coupled geodesic active regions for image segmentation: a level set approach. *Eur Conf Comput Vision* 224–240
- Xie X, Mirmehdi M (2008) MAC: magnetostatic active contour model. *IEEE Trans Pattern Anal Mach Intell* 30(4):632–646
- Huang X, Metaxas D (2008) Metamorphs: deformable shape and appearance models. *IEEE Trans Patt Anal Mach Intell* 30(8):1444–1459
- Adams R, Bischof L (1994) Seeded region growing. *IEEE Trans Pattern Anal Mach Intell* 16(6):641–647
- Fung PW, Grebbin G, Attikiouzel Y (1990) Model-based region growing segmentation of textured images. *IEEE Int Conf Acoust Speech Signal Process* 4:2313–2316
- Xiaohan Y, Yla-Jailski J, Huttunen O, Vehkomiiki T, Sipila O, Katila T (1992) Image segmentation combining region growing and edge detection. *Int Conf Pattern Recognit* 3:481–484
- Yu Q, Clausi DA (2008) IRGS: image segmentation using edge penalties and region growing. *IEEE Trans Pattern Anal Mach Intell* 30(12):2126–2139
- Qin AK, Clausi DA (2010) Multivariate image segmentation using semantic region growing with adaptive edge penalty. *IEEE Trans Image Process* 19(8):2157–2170
- Perona P, Malik J (1990) Scale-space and edge detection using anisotropic diffusion. *IEEE Trans Pattern Anal Mach Intell* 22(8):629–639
- Manay S, Yezzi A (2003) Anti-geometric diffusion for adaptive thresholding and fast segmentation. *IEEE Trans Image Process* 12(11):1310–1323
- Ji X, Feng J (2004) A new approach to thinning based on time-reversed heat conduction model. *IEEE Int Conf Image Process* 1:653–656

32. Kimia BB, Siddiqi K (1994) Geometric heat equation and non-linear diffusion of shapes and images. *IEEE Int Conf Comput Vision Pattern Recognit* 113–120
33. Makrogiannis SK, Bourbakis NG (2004) Motion analysis with application to assistive vision technology. *IEEE Int Conf Tools Artif Intell* 344–352
34. Direkoğlu C, Nixon MS (2006) Low level moving-feature extraction via heat flow analogy. *Int Symp Visual Comput LNCS* 4291:243–252
35. Direkoğlu C, Nixon MS (2010) Moving-edge detection via heat flow analogy. *Pattern Recognit Lett* (in press)
36. Direkoğlu C, Nixon MS (2007) Shape extraction via heat flow analogy. *Int Conf Adv Concepts Intell Vision LNCS* 4678:553–564
37. Nixon MS, Liu XU, Direkoğlu C, Hurley DJ (2011) On using physical analogies for feature and shape extraction in computer vision. *Comput J* 54(1):11–25
38. Holman JP (2002) *Heat transfer*, McGraw-Hill, 9th edn
39. Wasilewski M (2002) Active contours without edges matlab codes. <http://www.postulate.org/segmentation.php>
40. Li C, Kao C, Gore J, Ding Z (2008) Region-scalable fitting energy based active contour matlab codes. <http://www.engr.uconn.edu/~cmli/>
41. Xu C, Prince JL (1999) Gradient vector flow snake matlab codes. <http://iacl.ece.jhu.edu/projects/gvf/>