Computer Vision and Image Understanding 132 (2015) 39–55

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

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

Matching-constrained active contours with affine-invariant shape prior[☆]

Junyan Wang[a,b,*], Sai-Kit Yeung[b], Kap Luk Chan[a]

[a] School of Electrical & Electronic Engineering, Nanyang Technological University, Block S2.1, 50 Nanyang Avenue, Singapore 639798, Singapore
[b] Singapore University of Technology and Design, Singapore 138682, Singapore

ARTICLE INFO

Article history:
Received 19 November 2013
Accepted 13 November 2014
Available online 20 November 2014

Keywords:
Automatic object segmentation
Active contour
Object matching
Affine invariance
Matching-constrained active contour
Interior-points-to-shape relation

ABSTRACT

In the object segmentation by active contours, an initial contour provided by user is often required. This paper extends the conventional active contour model by incorporating feature matching in the formulation for automatic object segmentation, yielding a novel matching-constrained active contour. The key to our formulation is a mathematical model of the relationship between interior feature points and object shape, called the interior-points-to-shape relation. According to this interior-points-to-shape relation, we are able to achieve the automatic object segmentation in two steps. Specifically, we are able to estimate the object boundary position given the matched interior feature points. Afterwards, we are able to further optimize the boundary position in the active contour framework. To obtain a unified optimization model for this task, we additionally formulate the matching score as a constraint to active contour model, resulting in our matching-constrained active contour. We also derive the projected-gradient descent equations to solve the constrained optimization. In the experiments, we show that our method achieves automatic object segmentation, and it clearly outperforms the related methods.

1. Introduction

Automatic object segmentation is desirable in many higher level computer vision tasks. Active contour model is one of the most well-known models for object segmentation. However, it generally requires the user to provide a contour curve sufficiently close to the object boundary to initiate the segmentation. Hence, the existing segmentation frameworks of active contours are generally semi-automatic. In Fig. 1, we show how the initialization affects the result of conventional affine shape prior based active contour [6,40].

1.1. Related works

Relaxation of initialization for active contours. Relaxation of the user initialization in active contours for object segmentation is an emergent research topic [43,33,25,42,41]. Previous work, e.g. [23], for automatic initialization of the edge-based active contours selects the initial contours that gives approximately the minimum energy. In other words, it is assumed that most of the edges in the image are of interest so that all of those should be considered in the optimization. This assumption is valid for the images studied in [23] but it cannot be generalized.

Global optimization of active contour with shape priors for automatic object segmentation. Recently, active contour with shape priors have been popularized due to a series ground-breaking works, e.g. [8,22,35] to list a few. The global optimization techniques for some region-based and edge-based active contour models have been proposed in [7,37]. In [7], Cremers et al. proposed a branch-and-bound method for approximating the exhaustive search in a region-based active contour with shape prior modeling. Schoenemann and Cremers in [36,37] proposed a functional ratio energy for characterizing the object boundaries. The optimization is achieved via minimum ratio cycle algorithm by Lawler [21]. These global optimization methods do not require initial contour. Hence, these methods achieve automatic object segmentation. The idea behind these methods is to approximate the exhaustive search of the globally optimal contour curve over the entire image domain. These methods provide efficacious numerical solutions to global optimization of many active contour models. However, it can be observed that sometimes the object of interest is one of the suboptimal solutions to the active contour models, as shown in the experiment section of this paper.

[☆] This paper has been recommended for acceptance Luminita Vese.

[*] Corresponding author at: Singapore University of Technology and Design, Singapore 138682, Singapore.

E-mail addresses: wa0009an@e.ntu.edu.sg (J. Wang), saikit@sutd.edu.sg (S.-K. Yeung), eklchan@ntu.edu.sg (K.L. Chan).

http://dx.doi.org/10.1016/j.cviu.2014.11.002
1077-3142/© 2014 Elsevier Inc. All rights reserved.

Prior works on constrained implicit curve evolution. A pioneering work on incorporating constraints in the level set based segmentation framework was [34], in which Paragios proposed the first effective way for leveraging sparse control points as user input in level set based segmentation framework. In his paper, control points, a.k.a. landmarks, were chosen by the user, and an implicit shape prior model was constructed from the control points. Our work differs from his in several aspects. First, Paragios' work provides a natural model of the relationship between the implicit contour and the points near it, whereas we model the relationship between implicit contour and the points inside the contour. Second, Paragios considered the user-input control points as a soft constraint formulated in the objective function. In contrast, we consider matching as a hard constraint because of the useful concept of *initial feasible solution* in constraint optimization which allows us to unify the two-step matching and segmentation framework into one single optimization model. Lastly, our framework requires an implicit shape representation other than the conventional level set representation, and the derivation of the gradient descent algorithm has to be different.

Affine invariance in active contours. Imposing affine invariance in active contour is an active research topic in recent years. Foulonneau et al. [12] proposed the first affine invariant active contour based on shape moments. It is a marvellous idea to measure affine invariance of any implicit contour using moments. Bryner et al. [2] also proposed to impose affine invariance in the shape analysis based on elastic Riemannian metric. More recently, and Bryner and Srivastava [1] proposed to further impose the affine invariance in the Bayesian active contour formulation. In this work, we adopt a different and simpler approach.

Relations between object matching and segmentation. Object matching based on locally discriminative feature matching automatically finds a set of points on the object of interest in the image [27,26,15] achieving coarse object localization. Nevertheless, the matching correspondences established by object matching does not necessarily defines the object boundary precisely.

1.2. Methodology

We address the automatic object segmentation by active contour following a divide-and-conquer strategy. We consider the problem of automatic object segmentation as a unity of two subproblems: the object detection and the boundary locating. Unlike the global search methodology in [7,37], the object detection can automatically provide the coarse localization of the object with a relatively high confidence. We expect to obtain the segmentation with the initialization provided by the object detector. The philosophy is that *if the object detection cannot be done properly, then the object segmentation is also hopeless*. We aim at extending the formulation of active contour models by using object detection. There exist many effective object detectors based on classification, e.g. the Haar-like feature based detectors [39] and the HOG descriptor based detectors [9] as well as object matching [27]. We consider object matching specifically for our task, since it can offer richer descriptions of the object, including its size and orientation.

1.3. Contributions

Our contributions of this paper mainly lie in three aspects.

1. We obtain a mathematical model of the relationship between the interior points and the shape contour. This model immediately gives rise to a two-step computational framework for automatic object segmentation. Based on this model, we are able to estimate the boundary contour given the matched interior points. Afterwards, the generated contour can be further optimized in the active contour framework because of it is in the form of implicit surface.
2. In addition, we formulate the automatic object segmentation with object matching and active contour as a novel constrained optimization model, which is called the matching-constrained active contour model. The initial contour generated by using the feature matching is the *initial feasible solution* to the constrained optimization.
3. We derive the projected-gradient descent equations for solving the constrained optimization.

Our approach may be considered conceptually similar to the frameworks of face feature detection [10]. In these frameworks, an initial guess of the face feature points is generated according to the face detection. Afterwards, the face feature points will be aligned to the face in the image. These frameworks were developed for faces rather than general objects. Besides, the integration of the matching and active contour to form one unified optimization model is numerically advantageous, while being rarely explored.

1.4. Organization

The rest of the paper is organized as follows. In Section 2, we describe the computational framework for automatic object segmentation based on object matching and active contour. In Section 3, we introduce our mathematical model of the affine-invariant interior-points-to-shape relation. In Section 4, we present our unified constrained optimization model of the framework, namely the matching-constrained active contour, and the projected gradient descent algorithm. In Section 5, we evaluate our method for automatic object segmentation on real images of cluttered scenes, we also compare our method with the state-of-the-art methods on example based object segmentation and we present the associated quantitative analysis with discussions. We conclude the paper with discussions in Section 6.

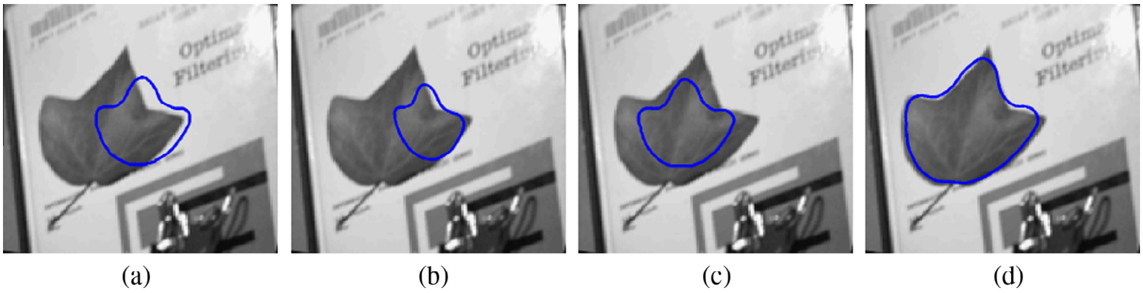


Fig. 1. The importance of initialization. (a) Bad initialization which gives an unsatisfactory segmentation result in (b) (see the details in text). (c) Good initialization which gives an encouraging result in (d).

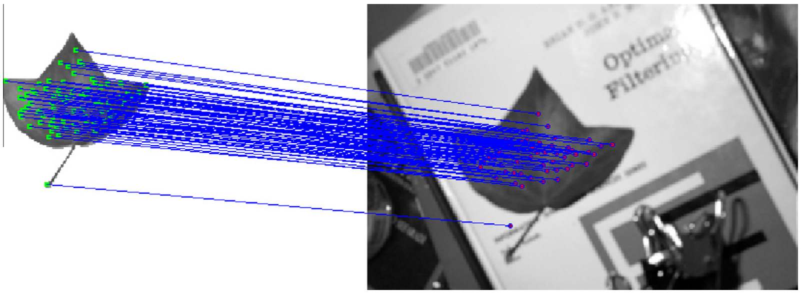


Fig. 2. Illustration of the feature point matching (with errors).

2. A computational framework for automatic object segmentation based on object matching and active contour

In this section, we describe a computational framework for automatic object segmentation based on object matching and active contour.

In object matching, the feature point matching often begins with an interest point detection on a pair of images as in [27]. Then the similarity between all possible pairs of matches are calculated. Finally, the matching algorithm will assign a set of feature points in one image, possibly a subset of the entire feature set, to a set of feature points in the other image according to the point-wise similarity. A correspondence is often represented by using binary variable, i.e. 1 means matched 0 means unmatched. In our paper, we will consider one image as an object template, and the task is known as object matching. The principle of the object matching is illustrated in Fig. 2, where the dots in two images are the feature points, and the lines across the images visualize the matching correspondences.

With the point correspondences established by the matching, we are able to solve the parameters of the transformation from the template object to the object in the image, and we are able to generate a contour in the image by, for example, applying the same transformation to the boundary contour of the object in the template. Ideally, we may use this contour to initialize the active contour. An example of the active contour given such a contour is shown in Fig. 3.

To construct this straightforward framework, we will have to overcome a major technical difficulty in this work, i.e. finding a contour representation, such that

1. the initial contour provided by matching is sufficiently close to the object boundary in the image, in order to facilitate the object segmentation by active contour, and
2. the contour provided by matching can be further optimized in the active contour framework.

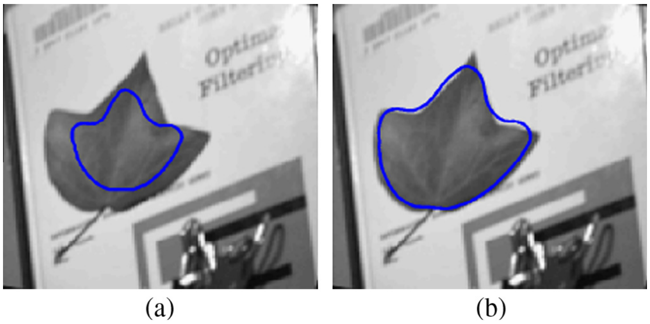


Fig. 3. Active contour with an initial contour. (a) Initial contour. (b) Converged solution.

To meet the former requirement, a contour representation which can be transformed according to the transformation of interior feature points will be needed. We call such contour representation as a transformation-invariant contour representation. Regarding the latter requirement, the implicit surface, which is known as a geometric (implicit) contour representation, is often adopted in the active contour framework. The argument on the geometric representation against the parametric representation can be found in the literature of active contours [4,5]. To summarize, we require *an implicit contour representation which can be transformed according to the transformation on the interior feature points.*

3. An implicit transformation-invariant contour representation

In this section, we deal with the aforementioned major technical difficulty. Specifically, we will propose an implicit contour representation which is determined by the positions of interior points. In addition, we impose affine-invariance on the proposed implicit contour representation to fulfill our requirements.

3.1. An implicit contour representation parameterized by positions of interior points

In this subsection, we propose an implicit contour representation determined by the positions of interior points, such that the contour representation can be sufficiently accurate and it changes according to the positions of interior points.

The shape of the template object can be represented by its silhouette. The shape silhouette of an object can be defined as a binary function $H_o(x,y)$ as follows:

$$H_o(x,y) = \begin{cases} 1, & [x,y] \in \Omega, \\ 0, & [x,y] \in \overline{\Omega}, \end{cases} \tag{1}$$

where Ω is the object region and $\overline{\Omega}$ is the non-object region. This definition can be used to define the object boundary contour of the object shape, denoted as C, which is written as follows:

$$C = \{x,y | \|\nabla H_o(x,y)\| \neq 0\}. \tag{2}$$

At the first glance of the above formulation, we would expect various ways for constructing this object region indication function H_o. For example, we can define $H_o = H(\phi(x,y))$, where H is a Heaviside function and ϕ is a signed distance function, and it is therefore closely related to the so-called *implicit contour representation* popularized in the last decade in the literature of level set method based segmentation [31,5,32].

To find the optimal contour representation we propose to adopt the following functional optimization problem:

$$\begin{aligned} H_e^* &= \underset{H_e \in \mathcal{S}}{\operatorname{argmin}}\ \mathcal{E}(H_e), \\ \mathcal{E}(H_e) &= \int_{\mathcal{D}} |H_o - H_e|^2 \, dx\, dy, \end{aligned} \tag{3}$$

where $\mathcal{D}$ is the entire image domain, H_e is the approximate of H_o, where H_o is the black-white object shape silhouette in the template, and $\mathcal{S}$ denotes the solution space of H_e.

In our context, we require H_e to be determined by the set of the position vectors of feature points $\{\vec{\mathbf{p}}\} = \{\vec{p}_1, \vec{p}_2, \ldots, \vec{p}_N\}$, i.e. $H_e = H_e(\{\vec{p}\})$. This H_e represents the *interior-points-to-shape relation.*

If we consider the feature points $\{\vec{\mathbf{p}}\}$ as randomly distributed points, we may adopt the radial basis functions (RBF) to construct H_e:

$$H_e(\vec{z}) = H\left(\sum_{i=1}^{N} \alpha_i \psi(\|\vec{z} - \vec{p}_i\|) + \beta\right), \quad (4)$$

where $\vec{z} = [x, y]^T$, H is still the Heaviside function, and we may call

$$\phi(\vec{z}) = \sum_{i=1}^{N} \alpha_i \psi(\|\vec{z} - \vec{p}_i\|) + \beta \quad (5)$$

the *shape decision function* in which $\psi(\cdot)$ is a kernel function, $\{\alpha_i\}$, β are the weights and bias to be determined. The points $\{\vec{\mathbf{p}}\}$ are the center points of the kernel functions. The sign of shape decision function determines whether a point belongs to the shape. An example of such interior-points-to-shape relation is shown in Fig. 4.

The usefulness of the formulation of contour representation in the above form has been explored in the literature of active contour, such as [13,29,38,30] to list a few. This formulation has been used to cope with noisy data [13], it can also be used to adapt to topology changes in a pure parametric active contour framework [29], and it can lead to efficient implementations [38] and even more flexible formulations of the segmentation [38]. In this work, we revisit the same formulation for the new problem of relating interior feature points to the contour shape.

Combining the previous formulations in Eqs. (1), (3) and (4), we obtain a binary classification problem based on RBF neural network in which the decision function is formed by only some of the positive samples, i.e. the feature points, but the training is accomplished by using both positive and negative samples over the entire image domain. We consider that the training is done by direct minimization of the fitting error in (3) with respect to the parameters $\{\alpha_i\}$ and β. The gradient descent equations for learning the parameters are given in Appendix A. The detailed learning strategy is presented in the section of experiment.

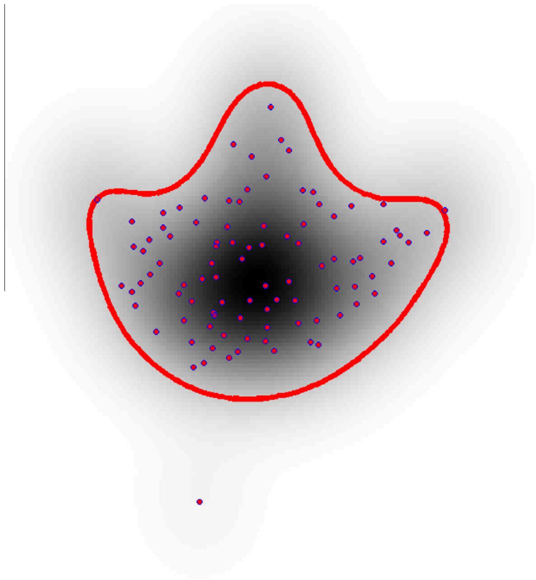


Fig. 4. A 2D visualization of the trained shape decision function overlayed the corresponding implicit contour (in red). The larger the intensity of the image is the more likely the point would belong to the shape region. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3.2. Imposing affine invariance on the interior-points-to-shape relation

The model defined in Eqs. (1), (3) and (4) gives us an implicit contour representation determined by the positions of interior points, which can also be sufficiently accurate. In addition, in our problem, the interior points will likely be transformed across the template and the image. Therefore, we require the interior points and the implicit contour to undergo the same transformation, i.e. being transformation-invariant.

The need for affine invariance in interior-points-to-shape relation. In this work, we consider a specific yet common type of transformation, i.e. the affine transformation, and the affine invariance property is also illustrated in Fig. 5. In the following, we first verify if the interior-points-to-shape relation obtained in the last subsection possesses the transformation invariance property.

Let us consider the model proposed in last subsection in Eq. (4). We choose the Gaussian function as the RBF kernel. Thus, the shape decision function can be written as follows.

$$\phi(\vec{z}) = \sum_{i=1}^{N} \alpha_i \exp\left(-\frac{\|\vec{z} - \vec{p}_i\|^2}{\sigma^2}\right) + \beta, \quad (6)$$

where σ is to be determined beforehand.

The contour curves can be defined by the shape decision function according to $C = \{x, y | \phi(x, y) = 0\}$, which is equivalent to Eq. (2) where H_o is replaced by H_e.

Let us consider the affine transformation of the kernel centers $\vec{p}_i$ as follows:

$$\vec{q}_i = \mathbf{A}\vec{p}_i + \vec{b}, \quad \text{for all } i, \quad (7)$$

where $\mathbf{A}$ is an invertible matrix, $\vec{b}$ is a translation vector. The corresponding shape decision function in terms of the transformed kernel centers can be written as follows:

$$\phi(\vec{z}, \{\mathbf{A}, \vec{b}\}) = \sum_{i=1}^{N} \alpha_i \exp\left(-\frac{\|\vec{z} - \mathbf{A}\vec{p}_i - \vec{b}\|^2}{\sigma^2}\right) + \beta. \quad (8)$$

The affine transformation of the contour points can be represented by $\vec{z}_s = \mathbf{A}\vec{z}_c + \vec{b}$, where $\vec{z}_c \in C$, such that $\phi(\vec{z}_c) = 0$ in (6). By substituting $\vec{z}_s$ into Eq. (8), we obtain the following:

$$\phi(\vec{z}_s, \{\mathbf{A}, \vec{b}\}) = \sum_{i=1}^{N} \alpha_i \exp\left(-\frac{\|\mathbf{A}\vec{z}_c - \mathbf{A}\vec{p}_i\|^2}{\sigma^2}\right) + \beta.$$

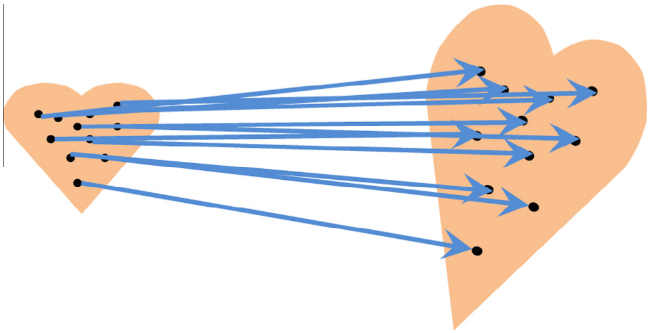


Fig. 5. Illustration of the need of affine invariance. The point correspondences should lead to the shape alignment.

Regarding the affine invariance property, the above leads to the following.

Proposition 3.1. *There exists infinitely many $\vec{z}_c$ and $\vec{p}_i$, such that for each i, we have* $\exp\left(-\frac{\|\mathbf{A}\vec{z}_c-\mathbf{A}\vec{p}_i\|^2}{\sigma^2}\right) \neq \exp\left(-\frac{\|\vec{z}_c-\vec{p}_i\|^2}{\sigma^2}\right)$, *if* $\mathbf{A}$ *is not an orthogonal matrix.*

The proof is deferred to Appendix B.

This means that the affine transformation of the contour curve defined by the trained shape decision function, $\phi(\vec{z})$, may not be the contour curve defined by the transformed shape decision function $\phi(\vec{z}, \{\mathbf{A}, \vec{b}\})$, which is an violation of the aforementioned affine invariance property.

Affine invariant interior-points-to-shape relation. To address the aforementioned problem, we propose a revised shape representation ϕ_S as follows.

$$\phi_S(\vec{z}, \{\mathbf{A}^{-1}, \vec{b}\}) = \sum_{i=1}^{N} \alpha_i \exp\left(\frac{-\|\mathbf{A}^{-1}(\vec{z} - \mathbf{A}\vec{p}_i - \vec{b})\|^2}{\sigma^2}\right) + \beta. \tag{9}$$

Substituting $\vec{z}_s$ into (9), we can verify that for general $\mathbf{A}$ and $\vec{b}$, we have the following affine invariance.

$$\phi_S(\vec{z}_s, \{\mathbf{A}^{-1}, \vec{b}\}) = \phi_S(\vec{z}_c, \{\mathbf{I}^{-1}, \vec{0}\}) = \phi(\vec{z}_c) = 0,$$

where $\mathbf{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ and $\vec{0}$ defines an identity transformation. We consider that ϕ_S is parameterized by $\mathbf{A}^{-1}$ rather than $\mathbf{A}$ to simplify the later derivation for optimizing $\mathbf{A}$.

We may further appreciate the significance of the affine invariance property by the example shown in Fig. 6. Suppose we are given the pair of initial points and shape in Fig. 4, the interior feature points of the leaf are then transformed by a predetermined affine transformation. The implicit contour curve defined by Eq. (8) is shown in Fig. 6(a). Note that Eq. (8) does not have the affine invariance property. The resultant shape, which tends to be a circle, differs significantly from a shape of a leaf. With the same transformation of the feature points, the implicit contour curves defined by the affine invariant shape decision function in Eq. (9) is shown in Fig. 6(b). Obviously, the affine-invariant implicit contour in Fig. 6(b) is like a leaf transformed resulting from the same transformation applied to the feature points, which is the affine-invariance property. With such an affine invariant interior-points-to-shape relation in Eq. (9), we can achieve the affine points-to-shape alignment. A result of the points-to-shape alignment is shown in Fig. 7 based on the feature matching presented in Fig. 2.

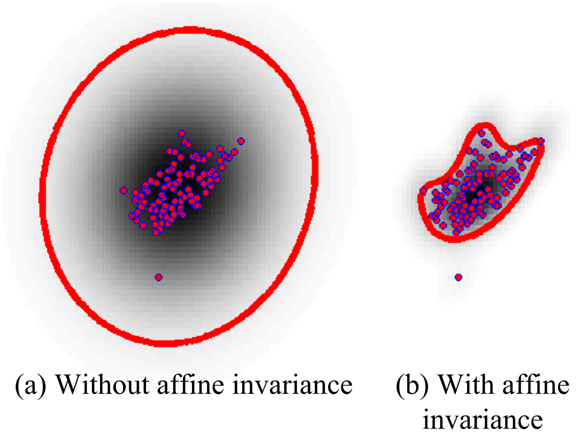

(a) Without affine invariance
(b) With affine invariance

Fig. 6. Significance of affine invariance of the shape representation. The dots are the transformed feature points. The outlining curves are the implicit shape contours.

4. A unified optimization model for automatic object segmentation

The established affine-invariant interior-points-to-shape relation completes our computational framework for automatic object segmentation. It also enables us to unify the framework into an optimization model. In this section, we propose our unified optimization model for automatic object segmentation.

4.1. Active contour model with a matching constraint

In this subsection, we propose a constrained optimization model to unify the object matching and active contour. The motivation of our constrained optimization model lies in the notion of *initial feasible solution*. Specifically, the contour curve given by the feature point matching provides the initial feasible solution to the constrained optimization model of active contour. The model is as follows:

$$\begin{aligned} &C^* = \underset{C}{\operatorname{argmin}}\; J(C(\{\vec{\mathbf{q}}\})), \\ &\text{s.t.}: \; E(\{\vec{\mathbf{q}}\}) \leqslant \tau, \quad \vec{q}_i = \mathbf{A}\vec{p}_i + \vec{b}, \end{aligned} \tag{10}$$

where $\{\vec{\mathbf{p}}\} = \{\vec{p}_1, \vec{p}_2, \ldots, \vec{p}_N\}$ and $\{\vec{\mathbf{q}}\} = \{\vec{q}_1, \vec{q}_2, \ldots, \vec{q}_N\}$ are the feature points on the template and the transformed sets of these feature points, $C(\{\vec{\mathbf{q}}\})$ denotes the implicit relation between the interior points and the contour shape defined by (9). $J(C)$ is the abstract form of the active contour energy, $E(\cdot)$ is the abstract form of the cost of joint matching and alignment, and τ is a tolerance level.

Note that the explicit contour optimization form of active contour model, as in Eq. (10), can often be written w.r.t. implicit contour representations [5,24,20]. In most of the cases, the two types of active contour models are equivalent. Without loss of generality, we choose to complete the paper with the explicit contour optimization form.

This formulation also ensures that the solved contour is close to the initial feasible contour. To find the initial feasible solution, it

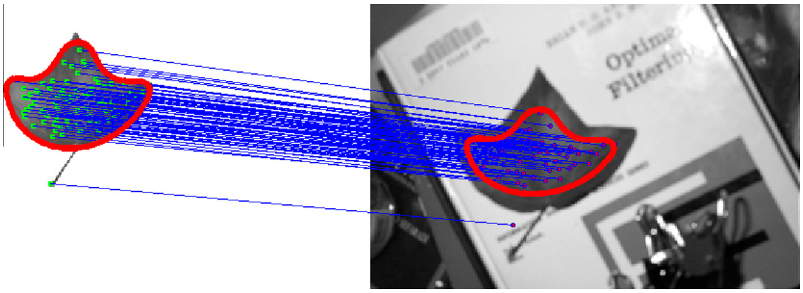


Fig. 7. An example of contour initialization based on feature matching.

suffices to minimize $E()$ w.r.t. $\{\vec{\mathbf{q}}\}$, since the set of $\{\vec{\mathbf{q}}\}$ that corresponds to the minimal value of $E(\cdot)$ must be feasible to the inequality constraint if there exists at least one feasible solution.

4.2. Joint formulation of point matching and alignment

In this subsection, we establish the explicit form of E in the constraint. Our formulation is a variant of the simplest form of the linear model of feature matching [14]. It can be written as follows:

$$\begin{aligned} &\{\vec{\mathbf{q}}^*\} = \underset{\{\vec{\mathbf{q}}\}}{\text{argmin}}\ E(\vec{\mathbf{q}}) = \sum_{ij} c_{ij}\delta_\epsilon(r_{ij}), \\ &\text{s.t. } \forall i,\ \sum_j \delta_\epsilon(r_{ij}) = 1, \end{aligned} \tag{11}$$

where c_{ij} is the so-called matching cost measuring the distance between all pairs of features across the two images. We use the SIFT interest point detector and the SIFT feature [27]. δ_ϵ is the approximation of the Dirac delta with a parameter $\epsilon, r_{ij} = \|\overrightarrow{q_i} - \overrightarrow{q_j^t}\|$ and $\{\overrightarrow{\mathbf{q}^t}\}$ is the set of target feature points in the target image, c_{ij} is the cost of the matching between $\overrightarrow{q_i}$ and $\overrightarrow{q_j^t}$. $\delta_\epsilon(r_{ij})$ is the relaxed matching indicator. $\delta_\epsilon(r_{ij})$ has the following property:

$$\delta_\epsilon(r_{ij}) \approx \begin{cases} M, & r_{ij} = 0, \\ 0, & r_{ij} \neq 0, \end{cases}$$

where M is a constant. In this model, the optimal δ_ϵ is determined by the closeness between $\overrightarrow{q_i}$ and $\overrightarrow{q_j^t}$. Thus, by aligning $\overrightarrow{q_i}$ toward a proper $\overrightarrow{q_j^t}$, the matching cost can be minimized. However, the above model is easily trapped by degenerate solutions where $\delta_\epsilon(r_{ij}) = 0$. To avoid such degeneration, we have the following reformulation:

$$\begin{aligned} &\{\vec{\mathbf{q}}^*\} = \underset{\{\vec{\mathbf{q}}\}}{\text{argmax}}\ \mathcal{E}(\{\vec{\mathbf{q}}\}) = \sum_{ij} e^{-c_{ij}}\delta_\epsilon(r_{ij}), \\ &\text{s.t. } \forall i,\ \sum_j \delta_\epsilon(r_{ij}) = 1. \end{aligned} \tag{12}$$

Note that $\mathcal{E} \geqslant 0$. The constraint is therefore $-\mathcal{E} \leqslant \tau$, where $E = -\mathcal{E}$.

4.3. Projected gradient decent algorithm

We are now able to derive the solution to the constrained optimization problem based on the idea of projected gradient descent method [28]. Projected gradient method tries to find the update direction of the current solution along which the cost function can be decreased and the constraints will not be violated. Since the constraints can often be expressed or transformed into equality constraints, the update direction is also required to lie within the *NULL* space of the constraints. While the notion of *NULL* space is mainly meant for linear equations, the nonlinear equality constraints can also be linearized, by e.g. Taylor expansion.

Our projected-gradient descent equations are as follows:

$$\frac{d\mathbf{A}}{dt} = -\nabla_{\mathbf{A}}J + \left\langle \nabla_{\mathbf{A}}J, \frac{\nabla_{\mathbf{A}}E}{\|\nabla_{\mathbf{A}}E\|} \right\rangle \frac{\nabla_{\mathbf{A}}E}{\|\nabla_{\mathbf{A}}E\|}, \tag{13}$$

$$\frac{d\vec{b}}{dt} = -\nabla_{\vec{b}}J + \left\langle \nabla_{\vec{b}}J, \frac{\nabla_{\vec{b}}E}{\|\nabla_{\vec{b}}E\|} \right\rangle \frac{\nabla_{\vec{b}}E}{\|\nabla_{\vec{b}}E\|}. \tag{14}$$

Note that these equations are not exactly the same as the ones in conventional projected gradient descent method in which inversion of a large matrix is needed. The rationale of Eqs. (13) and (14) lies in the following property.

$$\frac{dE}{dt} = 0 \quad \text{and} \quad \frac{dJ}{dt} \leqslant 0. \tag{15}$$

The derivation of this property is deferred to Appendix C. The above indicates that the projected gradient descent algorithm governed by Eqs. (13) and (14) can reduce J while leaving E unchanged. This might be too strong for the inequality constrained optimization. In fact, we only require E to be smaller than a predefined tolerance τ. Therefore, we implement the original gradient descent if $E < \tau$, and we implement the full projected gradient algorithm, if $E \approx \tau$.

To implement the projected gradient descent algorithm, we require the explicit form of the gradients $\nabla_{\mathbf{A}}J$. $\nabla_{\mathbf{A}}E, \nabla_{\vec{b}}J$ and $\nabla_{\vec{b}}E$. The derivation of the explicit form of $\nabla_{\mathbf{A}}J$ is complex. Alternatively, we can obtain $\nabla_{\mathbf{A}}J$ numerically by using $\nabla_{\mathbf{A}^{-1}}J$. The explicit form of $\nabla_{\mathbf{A}^{-1}}J$ and $\nabla_{\vec{b}}J$ can be written as follows:

$$\nabla_{\mathbf{A}^{-1}}J = -\left\langle \nabla J^T \overrightarrow{N}, \frac{\frac{D\phi_S}{D\mathbf{A}^{-1}}}{\langle \overrightarrow{N}, \nabla_{\vec{z}}\phi_S \rangle} \right\rangle_p, \tag{16}$$

$$\nabla_{\vec{b}}J = -\left\langle \nabla J^T \overrightarrow{N}, \frac{\frac{D\phi_S}{D\vec{b}}}{\langle \overrightarrow{N}, \nabla_{\vec{z}}\phi_S \rangle} \right\rangle_p, \tag{17}$$

in which $\overrightarrow{N}$ is the normal of the contour and $\nabla_{\vec{z}}\phi_S$ can be computed either numerically or exactly. The closed form expressions of $\frac{D\phi_S}{D\mathbf{A}^{-1}}$ and $\frac{D\phi_S}{D\vec{b}}$ are needed for the computation, which are as follows:

$$\begin{aligned} \frac{D\phi_S}{D\mathbf{A}^{-1}} &= -\sum_{i=1}^{N} 2w_i(\vec{z}) \frac{\vec{v}_i(\vec{z})\vec{z}^T}{\sigma^2}, \\ \frac{D\phi_S}{D\vec{b}} &= -\sum_{i=1}^{N} 2w_i(\vec{z}) \frac{\left(\mathbf{A}^{-1}\right)^T \vec{v}_i(\vec{z})}{\sigma^2}, \end{aligned}$$

where

$$\begin{aligned} w_i(\vec{z}) &= \alpha_i \exp\left(-\frac{\|\mathbf{A}^{-1}\vec{z} - \vec{p}_i - \mathbf{A}^{-1}\vec{b}\|^2}{\sigma^2} \right), \\ \vec{v}_i(\vec{z}) &= \mathbf{A}^{-1}\vec{z} - \vec{p}_i - \mathbf{A}^{-1}\vec{b}. \end{aligned}$$

The derivation is deferred to Appendix D. Our derivation is not restricted to any specific active contour model. For the Geodesic Active Contour (GAC) [3], which is a well-known edge-based active contour and can locate the object boundary accurately given a good initialization, the functional gradient, ∇J, is the following:

$$\nabla J_{GAC} = \langle \nabla g, \overrightarrow{N} \rangle \overrightarrow{N} - g\kappa\overrightarrow{N},$$

where g is an edge indicator function in which the stronger edge corresponds to smaller value, $\overrightarrow{N}$ is the normal of the contour, κ is the contour curvature.

The projected gradient algorithm also requires $\nabla_{\mathbf{A}}E$ and $\nabla_{\vec{b}}E$ which can be written as

$$\nabla_{\mathbf{A}}E\left(\mathbf{A}, \vec{b}\right) = -\sum_{ij} \hat{c}_{ij}g_{ij}\left(\mathbf{A}\vec{p}_i + \vec{b} - \overrightarrow{q_j^t}\right)\vec{p}_i^{\,T}, \tag{18}$$

$$\nabla_{\vec{b}}E\left(\mathbf{A}, \vec{b}\right) = -\sum_{ij} \hat{c}_{ij}g_{ij}\left(\mathbf{A}\vec{p}_i + \vec{b} - \overrightarrow{q_j^t}\right), \tag{19}$$

where $\hat{c}_{ij} = e^{-c_{ij}}$. We adopt the Gaussian function to approximate the Dirac delta. Thus, $g_{ij} = \delta_\epsilon(r_{ij})$. We also normalize the Gaussian functions according to the constraint $\sum_i \delta_\epsilon(r_{ij}) = 1$. These gradients may also be used for producing a local optimal solution to Eq. (12).

Given the above detailed formulations, we are able to derive our algorithm for the matching-constrained active contour. The pseudo code of our algorithm is presented in Algorithm 1. We also visualize the algorithm in a diagram in Fig. 8.

Algorithm 1. Projected gradient descent active contour.

```
Input  : Input image $I$, source points $\mathbf{P}$, matched target points $\mathbf{Q}$, trained $\phi_S(\cdot)$
Output: $H_e^*$
begin
    /*     Contour initialization by matching                    */
    Solving $[\mathbf{A}_o\ \vec{b}_o][\mathbf{P}\ \vec{1}]^T = \mathbf{Q}^t$ w.r.t. $\mathbf{A}_o, \vec{b}_o$
    repeat
        Update $\mathbf{A}_o, \vec{b}_o$ by Eqs. (18) and (19)
    until Convergence
    $\vec{p}_i^{\,t_0} \longleftarrow \mathbf{A}_o \vec{p}_i^{\,s} + \vec{b}_o$
    Evaluate $\phi_S(\vec{z}, \{\mathbf{A}_o^{-1}, \vec{b}_o\})$ on image domain by Eq. (9)
    /*     Constrained optimization                              */
    $k \longleftarrow 1$
    repeat
        if $E(\{\vec{p}_i^{\,t_{k-1}}\}) < \tau$ then
            Update $\mathbf{A}_k$ and $\vec{b}_k$ by applying direct gradient decent with the gradients
            defined in (16) and (17)
        else
            Update $\mathbf{A}_k$ and $\vec{b}_k$ by Eqs. (13) and (14)
        Evaluate $\phi_S(\vec{z}, \{\mathbf{A}_k^{-1}, \vec{b}_k\})$ image domain by Eq. (9)
        $k \longleftarrow k + 1$
    until Convergence
```

$\mathbf{P} = [\vec{p}_1, \vec{p}_2, ..., \vec{p}_N]^T$ is an $N \times 2$ matrix of feature vectors in the template image,

$\mathbf{Q}^t = [\vec{q}_1^{\,t}, \vec{q}_2^{\,t}, ..., \vec{q}_N^{\,t}]^T$ is an $N \times 2$ matrix of the corresponding feature vectors in the target image.

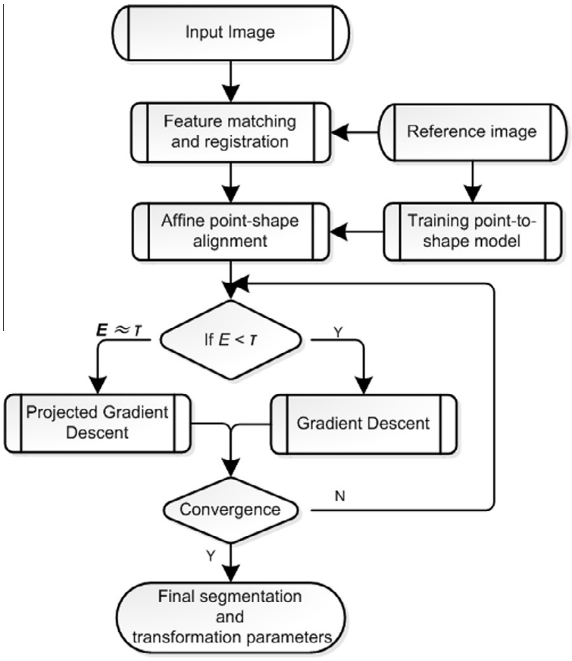

Input Image
Feature matching and registration
Reference image
Affine point-shape alignment
Training point-to-shape model
E ≈ T
If E < T
Y
Projected Gradient Descent
Gradient Descent
Convergence
N
Y
Final segmentation and transformation parameters

Fig. 8. Flow chat of matching-constrained active contour framework.

We adopt the locally affine matching [26] to provide the initial point correspondences in this work.

5. Experiments

In our implementation, we use the SIFT interest point detector and the SIFT feature [27]. The SIFT features are computed from the interest points detected in the image. On the template image, the SIFT points are extracted before the shape is cut out. We also used a distance map to select the feature points far enough from the boundary such that the feature values will not be dependent on the content outside the object region. To produce the initial matching, we adopt the locally affine matching for producing the initial point correspondences [26]. We are not confined to this choice of feature representation and matching algorithm. The τ in (10) is set in relation to the matching score E from initial object matching. More detailed discussions on the selection of τ are presented in Section 5.8.

We mainly experiment on the real images taken from Mikolajczyk's homepage,[1] Caltech computer vision archive[2] and the ETHZ Toys dataset to evaluate our method for automatic object segmentation.

[1] http://lear.inrialpes.fr/people/mikolajczyk/.

[2] http://www.vision.caltech.edu/html-files/archive.html.

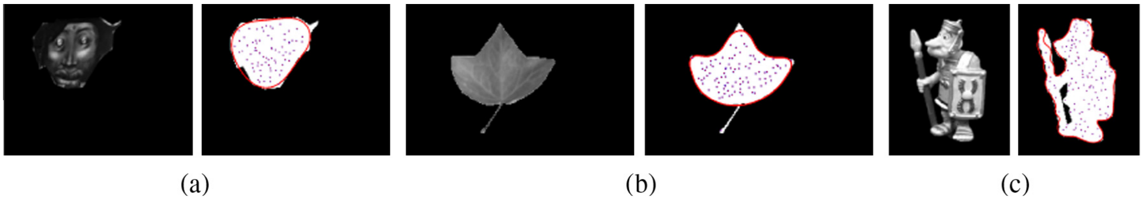


Fig. 9. Three pairs of template objects (left) and the corresponding trained shape contours overlaying the training shape (right).

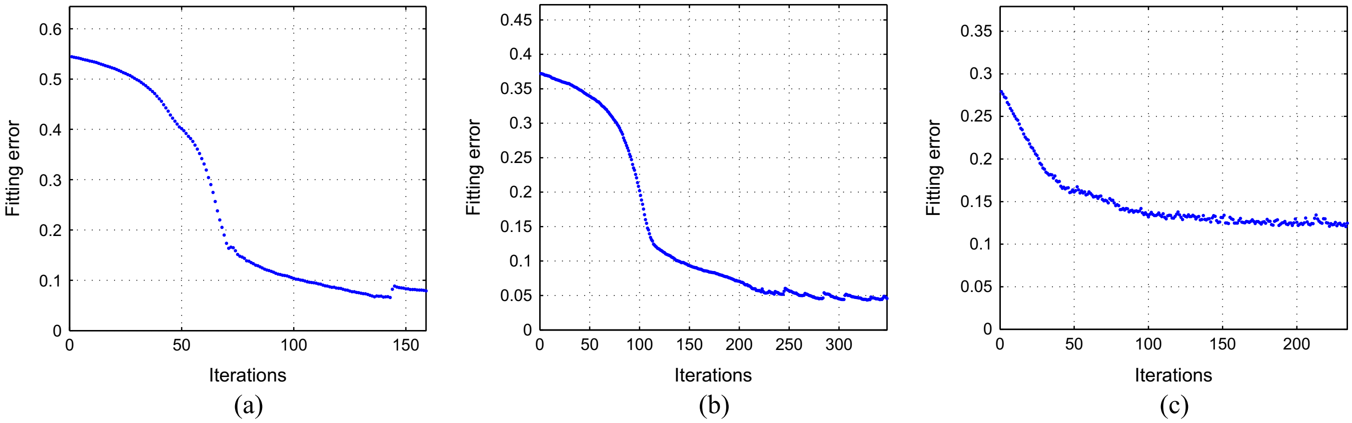


Fig. 10. Minimization of the errors of shape modeling corresponding to Fig. 9(a)–(c).

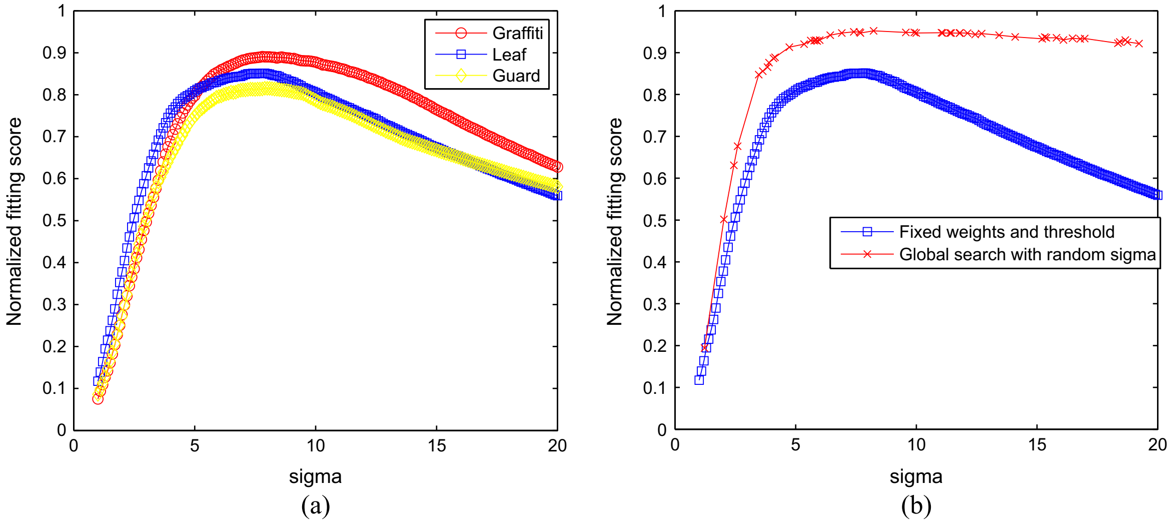


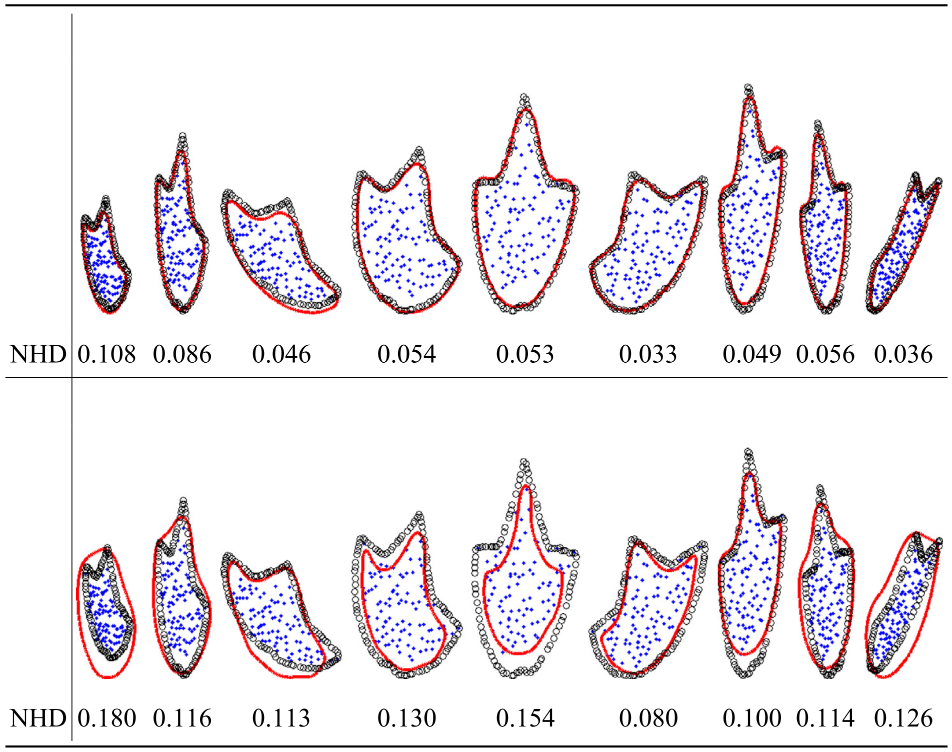
Fig. 11. Experimental results on selection of the parameter σ. (a) Plot of fitting scores (vertical) vs. σ (horizontal), given the fixed initial weights and thresholds. (b) Plot of fitting scores w.r.t. different values of σ given the fixed initial weights and thresholds (squares with curve) comparing with the fitting scores due to the optimization of the weights and threshold given the randomly selected σ (crosses with curve).

5.1. Affine invariant shape modeling

Following the shape training process presented in Section 3, we can obtain the affine invariant points-to-shape model. We present the template objects and the trained shape contours in Fig. 9. We also present the fitting errors during the training of the RBF for the shapes in Fig. 9. The training of the shape models converges stably (see Fig. 10).

There is a parameter σ in the RBF based shape representation. We may use the fitting errors with fixed initial weights α_i and threshold value β for selecting σ. Specifically, we select σ which corresponds to the highest fitting score ($1 - \textit{normalized error}$) from a set of $\{1, 1.1, \ldots, 19.9, 20\}$ containing 191 candidates. Fig. 11(a) shows the fitting scores w.r.t. σ for the three shapes. We also randomly select 50 values of σ from $[1, 20]$ and we implement the gradient descent learning to get the convergent shape models of the leaf shape corresponding to the 50 random values of σ. The scores of the optimal fitting, in terms of the Jaccard shape similarity, given the 50 random values of σ are shown in Fig. 11(b). We may observe that the peaks of the two curves are quite close, which means that the criteria for selecting σ with fixed weights and threshold is effective. The major benefit of the selection of σ before the shape model fitting is the computational efficiency.

Upon the accomplishment of the shape model training, we then validate the claimed affine invariance of the shape model defined in Eq. (9). We take the trained leaf shape model for evaluation.



NHD = Normalized Hausdorff distance

Fig. 12. The significance of the affine invariance of the interior-points-to-shape relation. The top row shows shapes generated by the affine invariant model, i.e. Eq. (9). The bottom row shows the shapes generated by the model without affine invariance, i.e. Eq. (8). The dots are the interior feature points, the curves are the implicit contours, and the circles are the explicitly transformed contours.

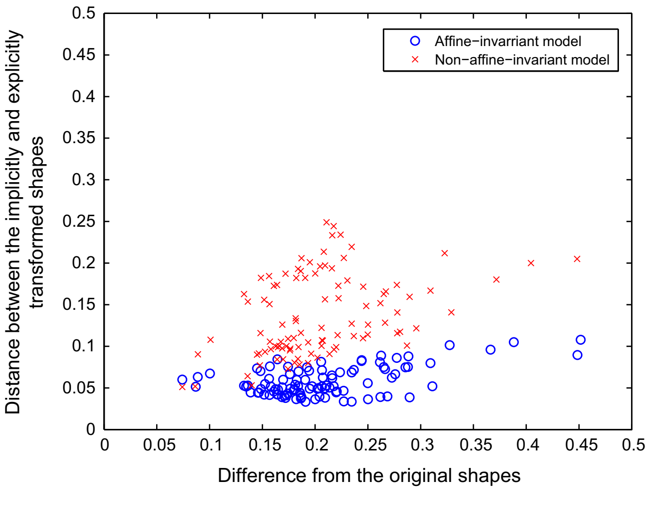


Fig. 13. The normalized Hausdorff distances between the shapes implicitly defined by the transformed interior points and the original shapes (horizontal) vs. the distance between the explicitly transformed contour shapes and the original shape (vertical).

The principle is to compare the contour shape implicitly defined by the transformed interior points with the explicitly transformed contour shape, which could be considered as the ground truth shapes. We choose the normalized Hausdorff distance, i.e. the Hausdorff distance normalized by the maximum distance between the two point sets, as the shape distance. We randomly generate the 100 sets of parameters for the affine transformation without translation, leading up to 100 randomly generated leaves. Some examples of the implicit contours overlaying with the explicitly transformed contour points are shown in the top row of Fig. 12. We can observe that the proposed interior-points-to-shape relation defined in Eq. (9) is affine invariant. In the bottom row of Fig. 12 we also present the implicit contours due to the shape model without affine invariance, i.e. Eq. (8). In Fig. 13, we plot the shape distances corresponding to the two shape models for all the 100 examples. We observe that even if the shape transformation is large, the distance between the explicitly and implicitly transformed contour shapes is still small for the affine-invariant shape model, whereas the shape model without the affine invariance deviates from the explicitly transformed shapes a lot. The explicitly transformed shapes can be considered as the ground-truth shapes.

5.2. Minimizing conventional active contour energies with exhaustive search

There could be various global optimization strategies for active contours such as those reported in [7,37]. In this subsection, we show that the global optimal solution to improper active contour energies may not correspond to the target object. This claim should not depend on the choice of the optimization method. Hence, we adopt the exhaustive search for the optimization. To examine the global optimality of object shape in image for a given active contour energy, we ensure that the object shape in the image has been included in the search space.

In this experiment, we perform exhaustive search for both the Chan–Vese model and the GAC model with hard but correct shape priors. In the implementation, we search over 8 orientations of the given shape, i.e. $\{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}, \pi, \frac{5\pi}{4}, \frac{3\pi}{2}, \frac{7\pi}{4}\}$, at all possible locations within the image domain. We make sure that this relatively small shape space covers roughly the correct object shape. A result is shown in Fig. 14. In this experiment, we have fixed the size of the shapes.

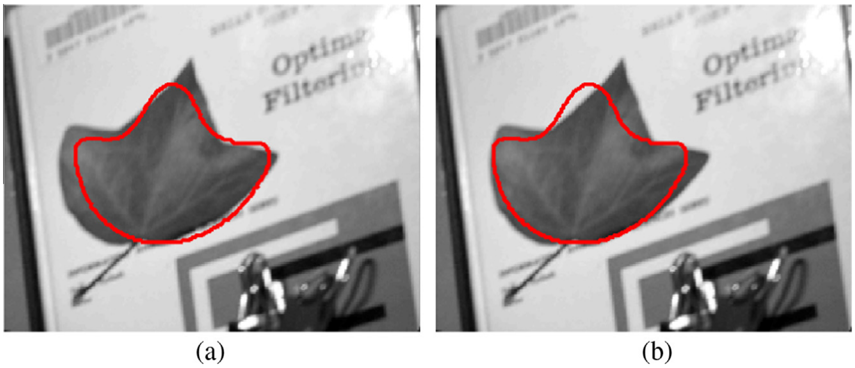


Fig. 14. The object segmentation by exhaustive search in simple image. (a) Result by global search with Chan–Vese model. (b) Result by global search with GAC model.

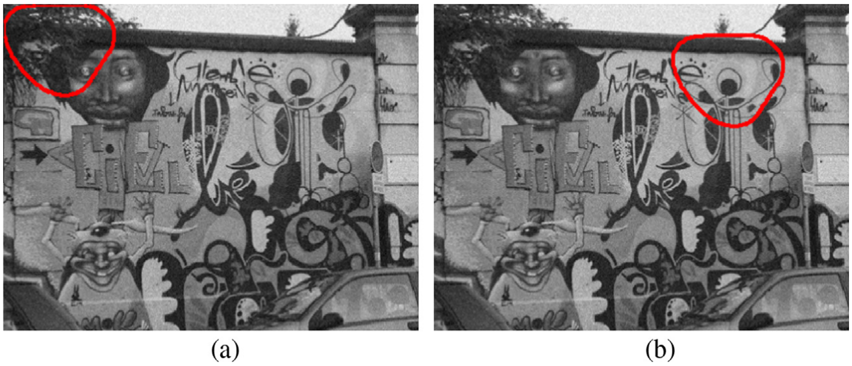


Fig. 15. The object segmentation by exhaustive search in a cluttered image. (a) Result by global search with Chan–Vese model. (b) Result by global search with GAC model.

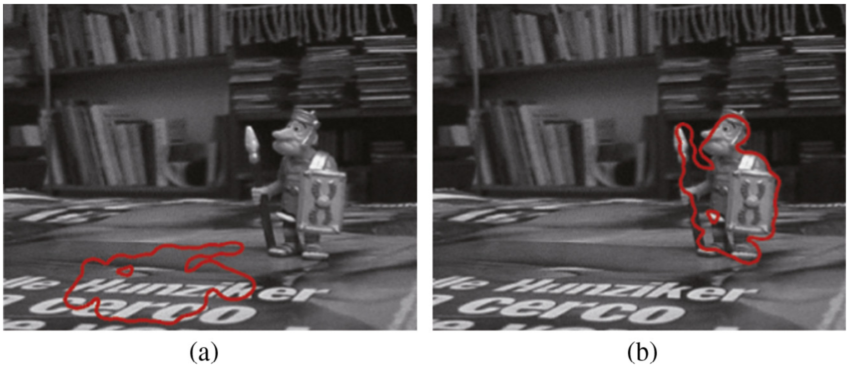


Fig. 16. The object segmentation by exhaustive search in a cluttered image containing single object, with the correct object size. (a) Result by global search with Chan–Vese model. (b) Result by global search with GAC model.

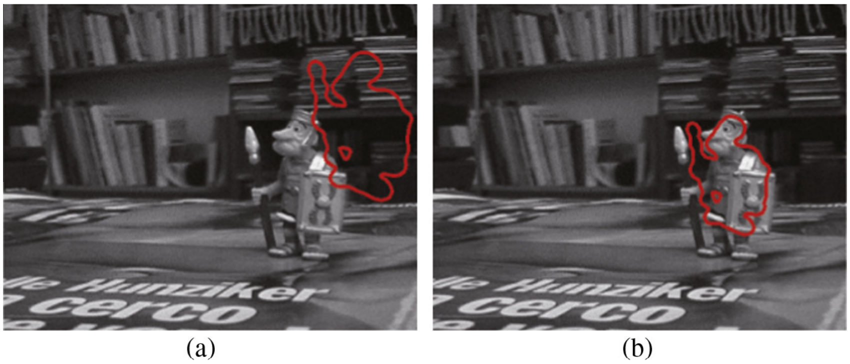


Fig. 17. The object segmentation by exhaustive search in a cluttered image containing single object, without knowing the size. (a) Result by global search with Chan–Vese model. (b) Result by global search with GAC model.

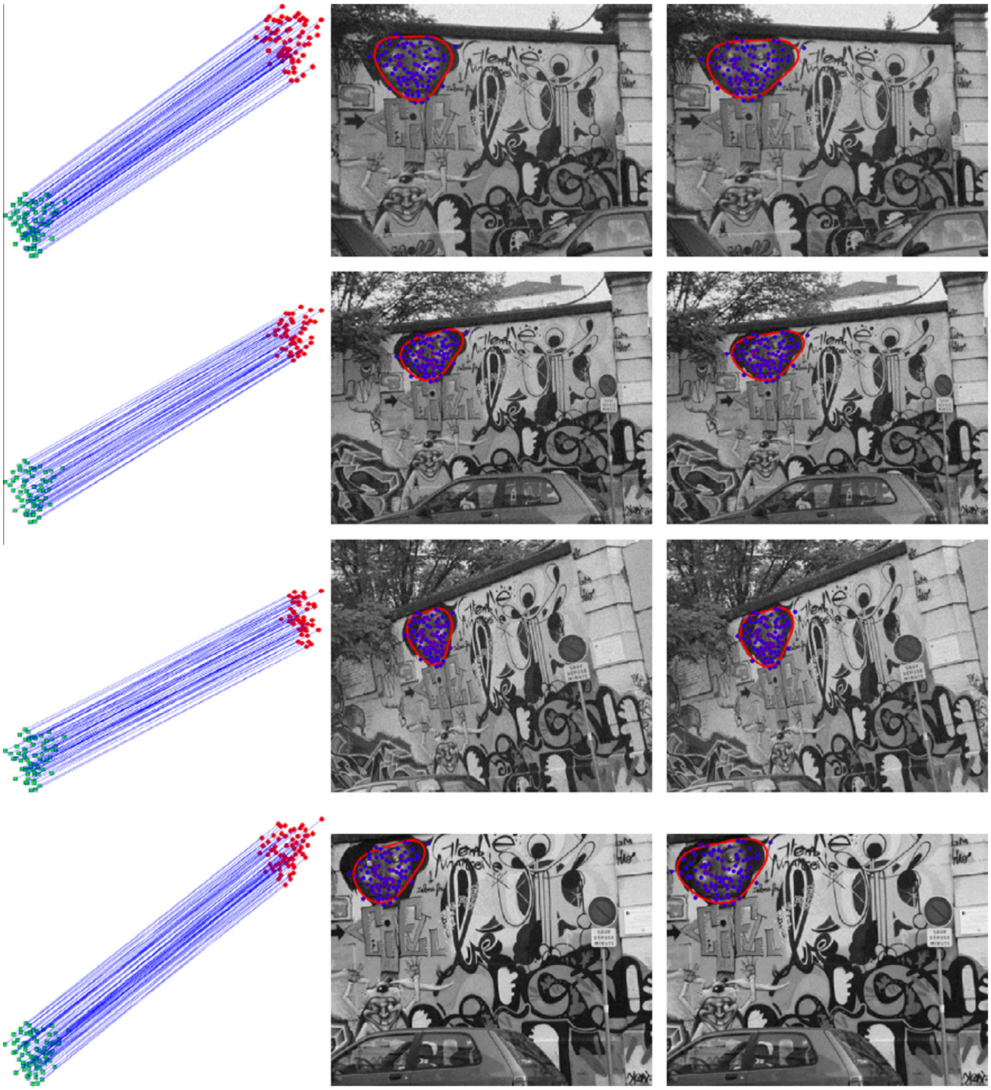


Fig. 18. Segmentation of an object of different poses in noisy images in the presence of occlusion by leaves, clutter background and indefinite boundary on the top. We label each triple of the figures from top to bottom as (a)–(d).

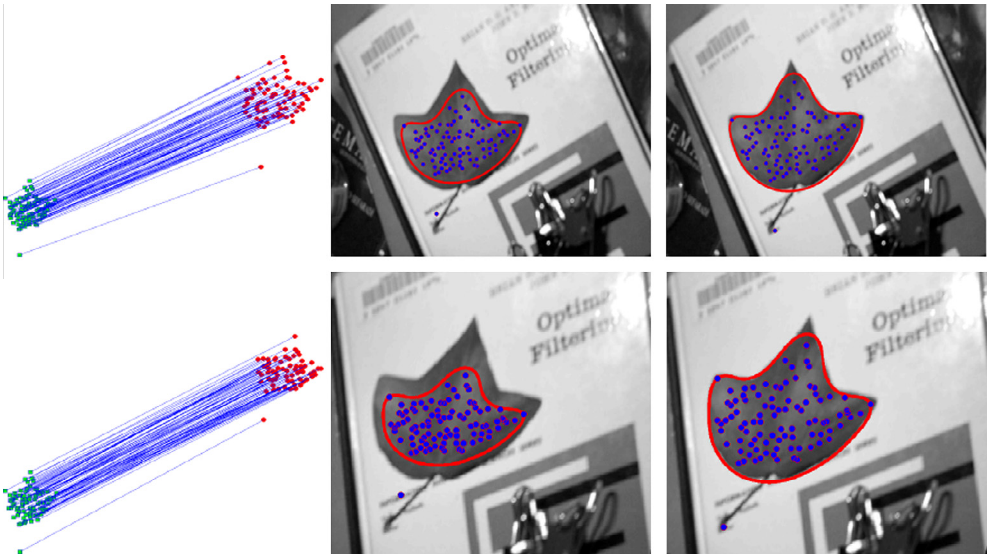


Fig. 19. Segmentation of an object of relatively complex shape under different poses and scales with non-ideal initial matchings. The top and bottom triple of figures are labeled as (a) and (b).

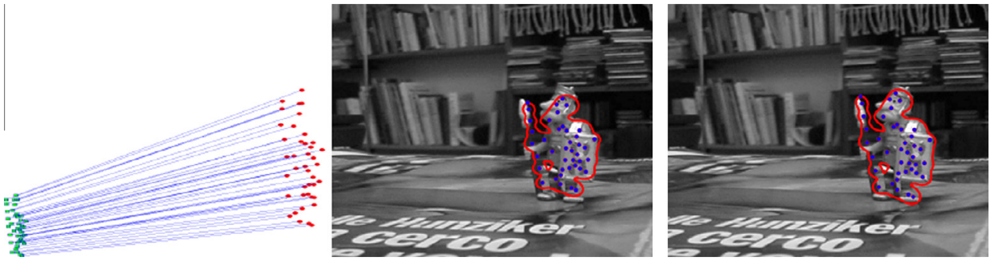


Fig. 20. Segmentation of an object of complex shape under different scale, clutter background and indefinite boundary.

Table 1
Jaccard similarity against the ground truth.

	Fig. 18				Fig. 19		Fig. 20
	(a)	(b)	(c)	(d)	(a)	(b)	
Initial	0.73	0.75	0.74	0.63	0.53	0.68	0.65
Final	0.75	0.84	0.86	0.86	0.90	0.93	0.75

We also show that given the correct fixed size, the global search also may not provide a reliable result of object segmentation in Fig. 15. We can observe that the global optimal solution to the Chan–Vese model (at the top-left corner) provides the region in which the image values contrast the outer region most, and the GAC model locates the group of the strongest edges that fit the shape prior best while not necessarily being the target object. Neither of the results are satisfactory. This is because we are not able to ensure the object of interest to correspond to the globally optimal solution of the formulated energy minimization problem.

There definitely exist cases that the global search can output satisfactory segmentation in relatively complex images given the correct size of the object. An example is shown in Fig. 16(b), which is a result of the global optimization for the GAC model. However, the global optimization of the Chan–Vese model is unsatisfactory, since the object of interest does not have significant contrast against the background. If we allow the size to vary in the search space, the results are often undesirable, such as in Fig. 17. In this experiment, we include the correct size with additionally one smaller and one larger sizes in the search space. The boundary of the template deviates a bit from the object boundary in the result of global optimization for GAC in Fig. 17(b), since the gradients on the object boundary are small while the gradients inside the object are significant. The result of segmentation by Chan–Vese model is in Fig. 17(a).

The experimental results in this subsection show that if the energy measure is improper, the global optimal solution of the active contour does not necessarily correspond to the desired object boundary.

5.3. Automatic object segmentation by matching-constrained active contour

In this subsection, we evaluate the matching-constrained active contour. The centroids of the template objects are set to be the

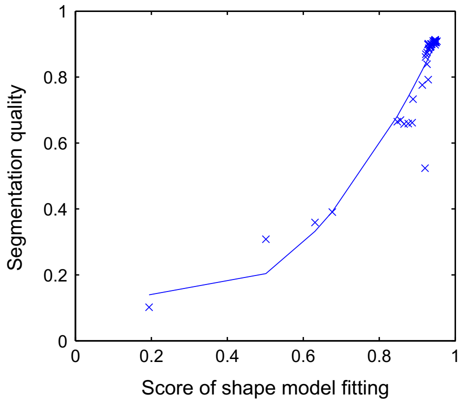


Fig. 21. Quality of segmentation (vertical) vs. score of model fitting (horizontal).

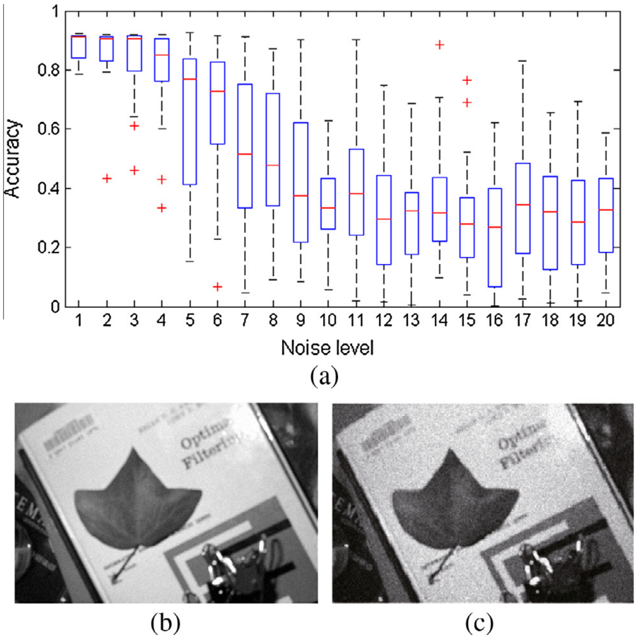


Fig. 22. Segmentation under different noise level. (a) Boxplot of the accuracy vs. noise level. (b) and (c) Images with noise. The standard deviation of the noise in (b) is 5, and the other is 20.

Table 2
Computational costs (seconds).

Figures	Fig. 18(a)	Fig. 18(b)	Fig. 18(c)	Fig. 18(d)	Fig. 19(a)	Fig. 19(b)	Fig. 20
Size(pixel)	320 × 400	320 × 400	320 × 400	320 × 400	296 × 448	282 × 448	411 × 408
M&R	17.83	18.3	18.53	17.9	18.44	16.27	10.05
GC	4.26	4.21	4.20	4.32	7.14	7.10	2.29
AC	98.14	40.93	82.03	650.56	504.97	401.95	82.95
Total	120.23	63.46	104.75	672.78	530.56	425.32	95.29

M&R = matching and registration, GC = generation of initial contour, AC = active contour before convergence, Total = total running time.

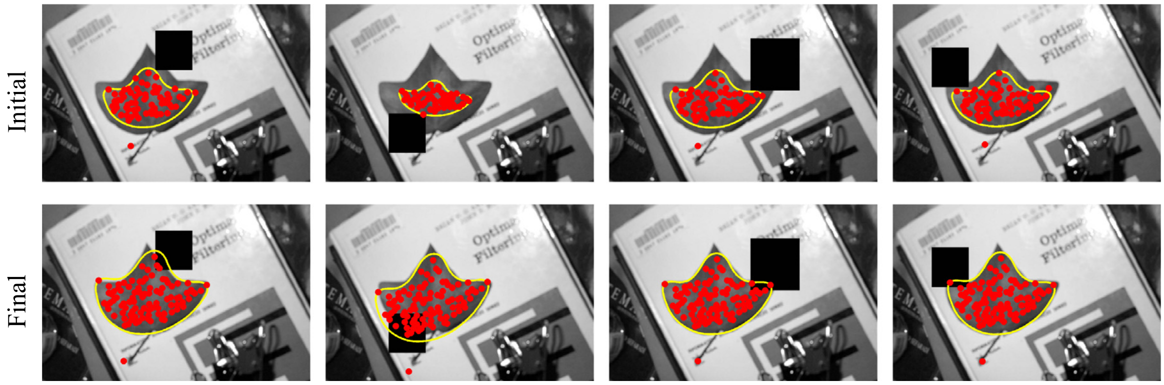


Fig. 23. Experiment for occlusion handling.

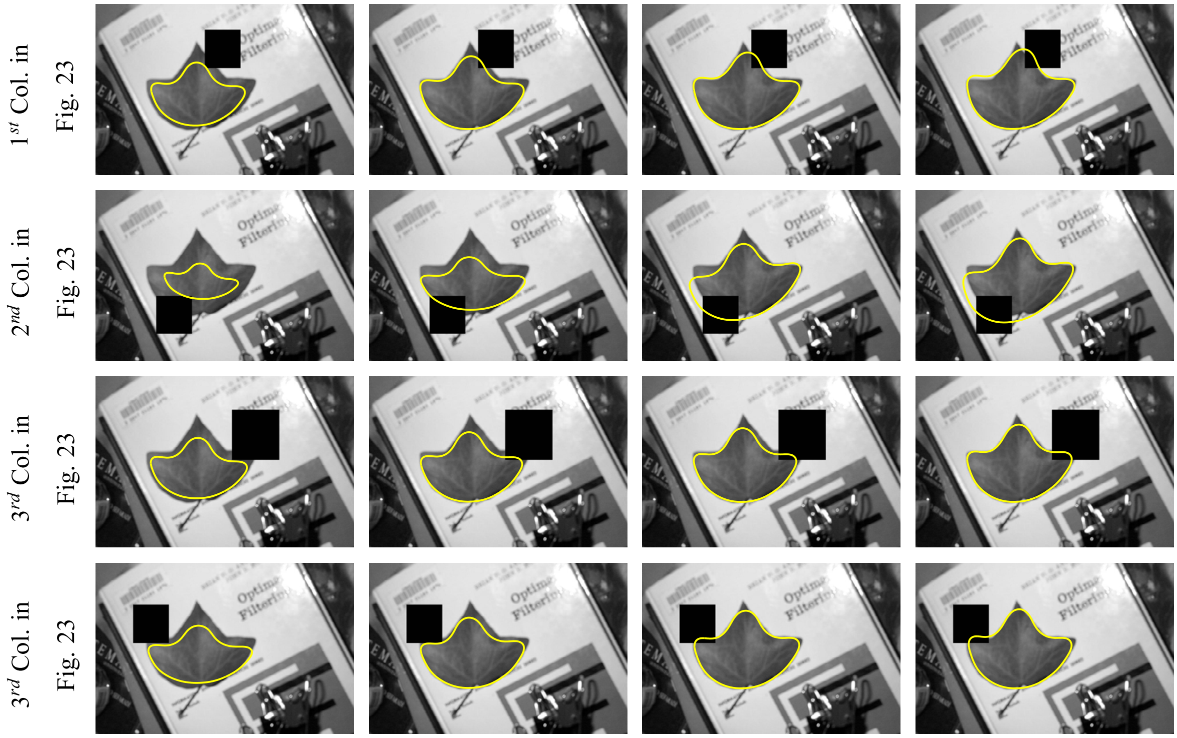


Fig. 24. Shape evolution processes for Fig. 23. Each row above, from left to right, shows the beginning to end of the shape evolution.

origin $(0,0)$. This position can be anywhere, and it does not affect the feature point matching since the formulation of feature point matching does not involve the absolute position.

From the point correspondences (obtained from SIFT matching) as shown in the left-most columns of Figs. 18–20, we obtain the initial contours shown in the middle columns of the figures and we finally obtain the segmentation results shown in the right-most columns of the figures. We can observe that the initial contours produced by our interior-points-to-shape relation are close to the object boundaries, and the active contour further improves the boundary location. Our method is capable of achieving the desirable segmentation results. We also compare the initial and final segmentation results with manually labeled ground truth shapes by Jaccard similarity measure to validate our visual observations quantitatively. We summarize the results in Table 1. We can see the improvement of the initial contour and the region enclosed by the final contour overlaps well with the ground truth. We also present the computational cost of the entire framework on the images in Table 2 for reference (Note that the entire process is run in MATLAB on a PC with Intel®Core™ i5-450M Processor and 4 GB memory).

We have implicitly assumed that a better shape model leads to a better result of segmentation throughout the paper. We ascertain this by experiments. We use the 50 shape models due to the 50 random σ generated previously for extraction of the leaf in the top image in Fig. 19. In Fig. 21, we show the strong correlation between the quality of shape modeling, in terms of fitting score, and the quality of segmentation, in terms of Jaccard similarity between the result and the ground truth.

5.4. Robustness to noise

We also evaluate our method under different noise level. We use the image in Fig. 19(a). We add Gaussian noise to the image. The means of the Gaussian noise are set to be zero and the standard deviations are varied from 1 to 20. For each noise level, we create 30 images and we apply our method to the images. The boxplot of the results of segmentation are shown in Fig. 22. We can observe that when the standard deviation of the noise is below 5, our method is robust. However, the decay of accuracy is sharp when the standard deviation is larger than 5. Hence, the proposed method may be sensitive to noise. This is because we adopt the SIFT feature for matching. This problem of sensitivity to noise in object matching have been addressed by robust feature representations such as the PCA-SIFT [18].

5.5. Object extraction under occlusion

We also study the performance of our method under occlusion. We create several cases of black box occlusions on the leaf of the leaf image. As shown in Fig. 23, while the initial solutions by matching are affected greatly by the occlusion, our method is still able to drive the contour to the correct boundary positions. However, this also means the matching algorithm that we adopted for producing the initial feasible solution is vulnerable to occlusion. A robust formulation of the matching constraint for handling partially observed objects is desirable. For the sake of clarity, we also present the shape evolution results in Fig. 24. We can observe that the affine-invariance also functions as a shape prior during the process.

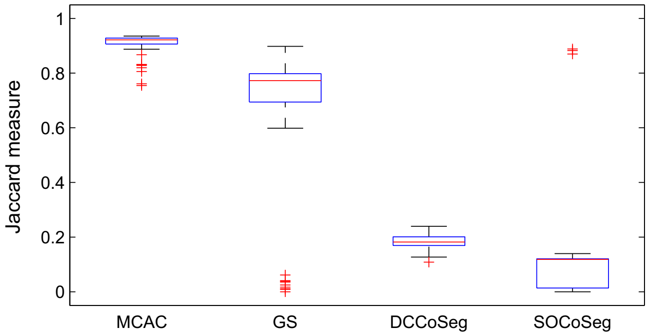


Fig. 25. Quantitative comparison.

5.6. Quantitative comparison to global optimization and co-segmentation

In this subsection, we conduct experiments for comparing our method with other related methods, including global optimization of the GAC model (global search, or GS), co-segmentation based on discriminative clustering (DCCoSeg) [17] and co-segmentation based on submodular optimization (SOCoSeg) [19]. We use eight orientations and 3 scales in the global search. The implementations of the co-segmentation algorithms are taken from the authors' websites. All the methods in the comparison require an object example. The task is also the same, namely to outline the same or similar object in the image of interest.

We conduct a quantitative experimental comparison. To eliminate miscellaneous factors in the experiment as many as possible, we use images generated by randomly transformation of the same image which is the leaf image shown in Fig. 14. We use 50 random affine transformation matrices to generate 50 random images. The object example has been shown in Fig. 9. The quantitative results are shown in Fig. 25. We can observe that our method, matching-constrained active contour (MCAC), significantly outperforms the others. Some visual results are also shown in Fig. 26. We can observe that our method can locate the object boundary satisfactorily. However, the results of global search may deviate from the object of interest because of insufficient information on the object. The co-segmentation methods may produce a lot of spurious contours and some times they may also miss the object. This is because the formulation of co-segmentation generally does not include a sensible object model. Our model integrates feature matching, shape prior modeling and object boundary modeling, all of which are practically useful object models.

5.7. Object extraction for image batch

In this experiment, we perform our method on an image sequence for capturing the same static object. Since we have not considered the temporal information in our model, we consider this experiment as an experiment for object extraction from image

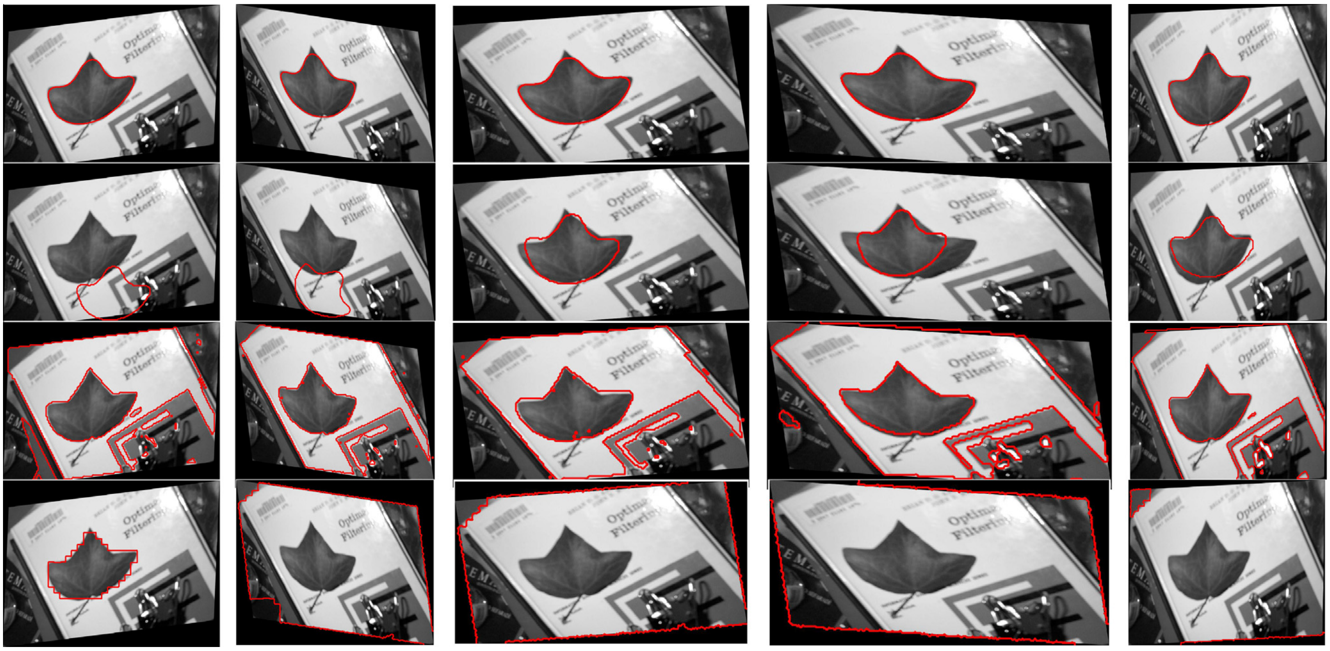


Fig. 26. Visual comparison. The first row to the 4th row shows the results of our method, global search, discriminative clustering based co-segmentation [17] and submodular optimization [19].

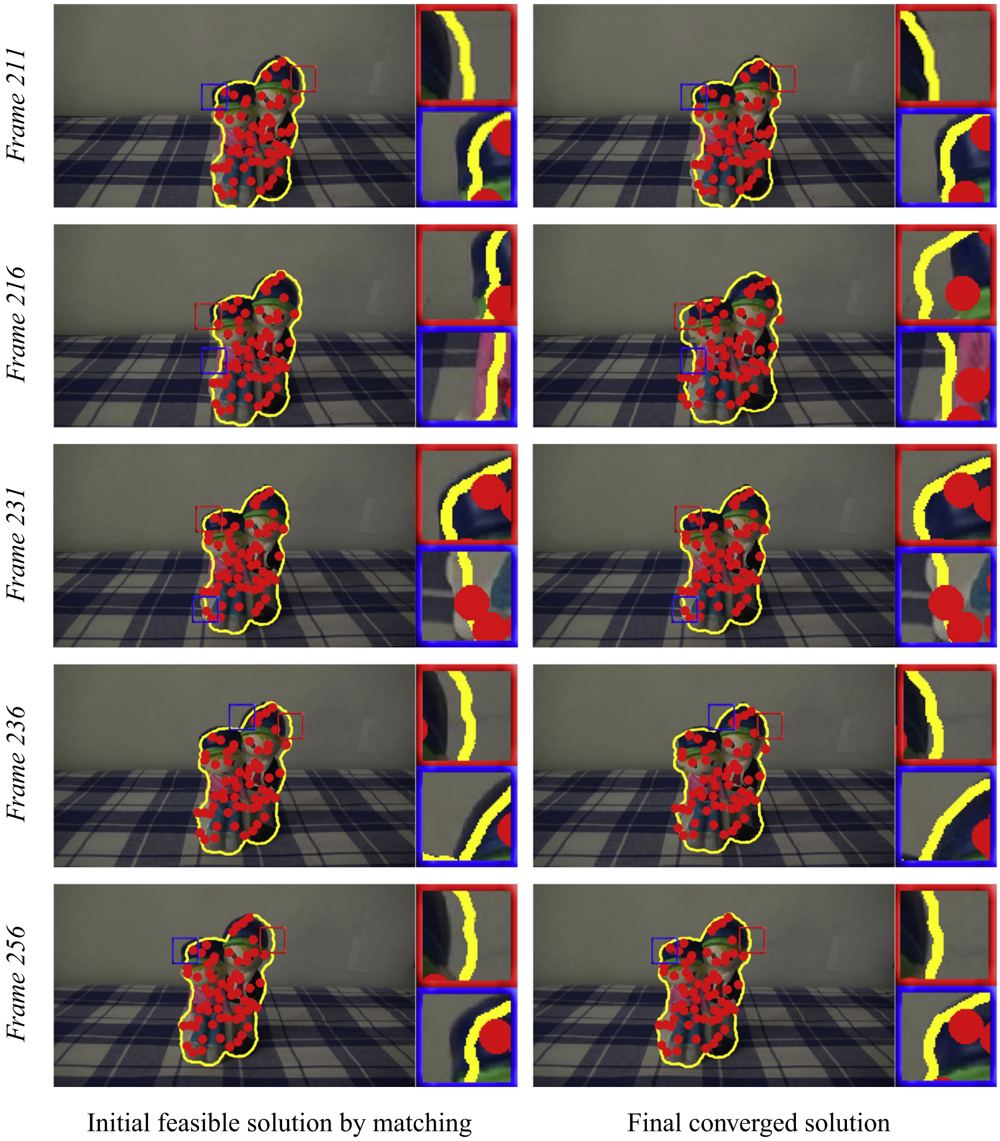


Fig. 27. Object extraction from image batch. In each row, the small figures on the right are the zoom-in areas of the boxed regions on the left.

batch. The shape model is trained on one template image. We used dense interior point in this experiment for training the shape decision function while we use only the SIFT points among them to form the matching constraint. As shown in Fig. 27, the initial feature matching can give reasonably good results while the boundary location was not precise, and our proposed projected gradient algorithm generally can further improve the segmentation. We can observe the result in the second row was not satisfactory. This is because the object does not undergoes the assumed affine transformation, but it undergoes a three dimensional perspective transformation. A 3D shape representation can be useful for handling this problem.

5.8. Discussions

There is a tolerance level τ in the formulation of the matching-constrained active contour to be determined by the user. We provide some observations to help the users to select their appropriate τ. Since the small τ may only allow small deviation from the initial solution even if the boundary is not very close to the initial contour. Hence, we may not need to constrain the optimization strictly. In the images for evaluation, the boundary of the objects of interest corresponds to the local optimal solutions to the active contour energy. Hence, we only need the gradient descent given the initial feasible solution produced by optimal matching without compromising to the constraint. The constraint should be more useful during the search for the optimal solution if the boundary is actually not a local optimal solution to the active contour model.

Objects may deform both rigidly and non-rigidly. In this paper, we have considered a general type of deformation, i.e. the affine transformation. The more general non-rigid deformations can be estimated by following the nonrigid shape prior modeling, e.g. [11,40,16], once the affine shape transformation is known.

Robust detection of the convergence of the gradient descent based method is a research problem in many areas such as optimization, machine learning. In our experiments, we terminate the algorithm if the average change of the affine parameters is less than a threshold or the maximum iteration is reached.

The computational cost depends on the number of selected feature points on the template object. A careful selection of the feature points can reduce the computational cost drastically while preserving the segmentation accuracy.

6. Conclusion and future work

In this paper, we proposed a novel constrained optimization formulation of active contours. The numerical optimization of this new active contour model leads to an automatic object segmentation algorithm. This work expands the capacity of the conventional active contour approach for object segmentation, and the extension has practical significance in that the conventional semi-automatic framework has been automated.

There are several possible future directions. (1) The shape modeling in our affine-invariant interior-point-to-shape relation can be improved, and the computations can be accelerated by cleverly selecting the feature points. (2) The object matching is based on single template. However, the appearance of the object of interest could vary significantly in different images. Robust object matching is crucial to our method. (3) There could be interesting applications of this method, such as in the training phase of general object recognition tasks. (4) We have not developed advanced shape prior model for this framework, and this is out of the scope of this work. We would like to consider the advanced shape prior modeling as a future work. (5) We have not considered inverse consistency which is known to be critical in many matching scenarios [44,45]. The inverse consistency and other matching priors can be incorporated in our model as additional constraints to achieve more accurate and robust segmentation.

Acknowledgments

Junyan Wang is supported by SUTD-ZJU Collaboration Research Grant 2012 SUTD-ZJU/RES/03/2012. Sai-Kit Yeung is supported by SUTD Start-Up Grant ISTD 2011 016, SUTD-MIT International Design Center Grant IDG31300106, and Singapore MOE Academic Research Fund MOE2013-T2-1-159.

Appendix A. Gradient descent equations for shape training

The gradient descent equations for learning the parameters $\{\alpha_i\}$ and β in the shape decision function in Eq. (4) are as follows:

$$\frac{\partial \alpha_i}{\partial t} = -\int_{\mathcal{D}} 2(H_e - H_o)\psi(\|\vec{z} - \vec{p}_i\|)H'\,dx\,dy,$$
$$i = 1, 2, \ldots, N,$$
$$\frac{\partial \beta}{\partial t} = -\int_{\mathcal{D}} 2(H_e - H_o)H'\,dx\,dy,$$

where H' is the first order derivative of H. The trained contour curve is defined by Eq. (2) in which H_o is replaced by the trained H_e.

Appendix B. Proof of Proposition 3.1

Proof. Let $\vec{a} = \vec{z}_c - \vec{p}_i$, we have that $\mathbf{A}\vec{a} = \mathbf{A}\vec{z}_c - \mathbf{A}\vec{p}_i$. Our objective is to show $\|\mathbf{A}\vec{a}\|^2 \neq \|\vec{a}\|^2$ in general. Since $\|\mathbf{A}\vec{a}\|^2 - \|\vec{a}\|^2 = \vec{a}^T(\mathbf{A}^T\mathbf{A} - I)\vec{a}$, and $\mathbf{A}^T\mathbf{A} - I$ is a nonzero symmetric matrix. We can write

$$\|\mathbf{A}\vec{a}\|^2 - \|\vec{a}\|^2 = \vec{b}^T D\vec{b},$$

where $\mathbf{A}^T\mathbf{A} - I = U^T DU$ and $\vec{b} = U\vec{a}$. This is a result of eigen-decomposition.

Obviously, there exists one vector $\vec{b}$ such that the above does not equal 0 for any nonzero $\mathbf{D}$. By almost arbitrary scaling of $\vec{b}$, we obtain infinitely many such vectors, which completes the proof. □

Appendix C. Derivations of Eq. (15)

In the following, we present the derivations of Eq. (15).

$$\begin{aligned}\frac{dE}{dt} &= -\nabla_{\mathbf{A}}E^T\left(\nabla_{\mathbf{A}}J - \left\langle \nabla_{\mathbf{A}}J, \frac{\nabla_{\mathbf{A}}E}{\|\nabla_{\mathbf{A}}E\|}\right\rangle \frac{\nabla_{\mathbf{A}}E}{\|\nabla_{\mathbf{A}}E\|}\right) \\ &\quad - \nabla_{\vec{b}}E^T\left(\nabla_{\vec{b}}J - \left\langle \nabla_{\vec{b}}J, \frac{\nabla_{\vec{b}}E}{\|\nabla_{\vec{b}}E\|}\right\rangle \frac{\nabla_{\vec{b}}E}{\|\nabla_{\vec{b}}E\|}\right) \\ &= -\langle \nabla_{\mathbf{A}}J, \nabla_{\mathbf{A}}E\rangle + \langle \nabla_{\mathbf{A}}J, \nabla_{\mathbf{A}}E\rangle - \langle \nabla_{\vec{b}}J, \nabla_{\vec{b}}E\rangle + \langle \nabla_{\vec{b}}J, \nabla_{\vec{b}}E\rangle = 0,\end{aligned}$$

$$\begin{aligned}\frac{dJ}{dt} &= -\nabla_{\mathbf{A}}J^T\left(\nabla_{\mathbf{A}}J - \left\langle \nabla_{\mathbf{A}}J, \frac{\nabla_{\mathbf{A}}E}{\|\nabla_{\mathbf{A}}E\|}\right\rangle \frac{\nabla_{\mathbf{A}}E}{\|\nabla_{\mathbf{A}}E\|}\right) \\ &\quad - \nabla_{\vec{b}}J^T\left(\nabla_{\vec{b}}J - \left\langle \nabla_{\vec{b}}J, \frac{\nabla_{\vec{b}}E}{\|\nabla_{\vec{b}}E\|}\right\rangle \frac{\nabla_{\vec{b}}E}{\|\nabla_{\vec{b}}E\|}\right) \\ &= -\left(\|\nabla_{\mathbf{A}}J\|^2 - \left\langle \nabla_{\mathbf{A}}J, \frac{\nabla_{\mathbf{A}}E}{\|\nabla_{\mathbf{A}}E\|}\right\rangle^2\right) \\ &\quad - \left(\|\nabla_{\vec{b}}J\|^2 - \left\langle \nabla_{\vec{b}}J, \frac{\nabla_{\vec{b}}E}{\|\nabla_{\vec{b}}E\|}\right\rangle^2\right) \leq 0.\end{aligned}$$

Appendix D. Derivation of Eqs. (16) and (17)

Our derivation is based on the following equality for minimizing general active contour energy.

$$\frac{\partial C}{\partial t} = -\nabla J = \alpha(p)\overrightarrow{N}. \tag{D-1}$$

Note that the equality holds true for geometric active contours.

The differential of the shape decision function ϕ_S at the implicit contour leads to the following.

$$\left.\frac{\partial \phi_S(\vec{z}(t), \{\mathbf{A}^{-1}(t), \vec{b}(t)\})}{\partial t}\right|_{\vec{z}\in\{\vec{z}|\phi_S=0\}} = \nabla_{\vec{z}}\phi_S \cdot \frac{\partial C}{\partial t} + \frac{D\phi_S}{\mathbf{DA}^{-1}} \cdot \frac{d\mathbf{A}^{-1}}{dt} + \frac{D\phi_S}{D\vec{b}} \cdot \frac{d\vec{b}}{dt} = 0, \tag{D-2}$$

where $\frac{\partial \phi_S}{\partial \vec{z}} = \nabla_{\vec{z}}\phi_S$, and $\frac{\partial C}{\partial t} = \frac{\partial \vec{z}}{\partial t}$. Note that C denotes the contour, and $\vec{z}$ is the point on the contour. In addition, the dot product above denotes elementwise inner product for both vectors and matrices.

Substituting Eq. (D-1) into the above, we obtain the expression for α.

$$\alpha = -\frac{\frac{D\phi_S}{\mathbf{DA}^{-1}} \cdot \frac{d\mathbf{A}^{-1}}{dt} + \frac{D\phi_S}{D\vec{b}} \cdot \frac{d\vec{b}}{dt}}{\langle \overrightarrow{N}, \nabla_{\vec{z}}\phi_S\rangle}. \tag{D-3}$$

Substituting the (D-3) into (D-1) we obtain the curve evolution as follows.

$$\frac{\partial C}{\partial t} = -\frac{\frac{D\phi_S}{\mathbf{DA}^{-1}} \cdot \frac{d\mathbf{A}^{-1}}{dt} + \frac{D\phi_S}{D\vec{b}} \cdot \frac{d\vec{b}}{dt}}{\langle \overrightarrow{N}, \nabla_{\vec{z}}\phi_S\rangle}\overrightarrow{N}. \tag{D-4}$$

To minimize a general active contour energy $J(C)$, we require the derivative of $J(C)$ to be non-positive as

$$\frac{dJ}{dt} = \left\langle \nabla J, \frac{\partial C}{\partial t}\right\rangle_p \leqslant 0, \tag{D-5}$$

where $\langle f, g\rangle_p = \int\langle f(p), g(p)\rangle dp$ is the inner product of two vector functions in which $\langle , \rangle$ is the vector inner product. Substituting (D-4) into (D-5), considering $\overrightarrow{N} = \nabla_{\vec{z}}\phi_S$, we obtain the following:

$$
\begin{aligned}
\frac{dJ}{dt} &= \left\langle \nabla J, -\frac{\frac{D\phi_S}{D\mathbf{A}^{-1}}\cdot\frac{d\mathbf{A}^{-1}}{dt}+\frac{D\phi_S}{D\vec{b}}\cdot\frac{d\vec{b}}{dt}}{\langle\overrightarrow{N},\nabla_{\vec{z}}\phi_S\rangle}\overrightarrow{N}\right\rangle_p \\
&= -\left\langle \nabla J^T\overrightarrow{N}, \frac{\frac{D\phi_S}{D\mathbf{A}^{-1}}\cdot\frac{d\mathbf{A}^{-1}}{dt}}{\langle\overrightarrow{N},\nabla_{\vec{z}}\phi_S\rangle}\right\rangle_p - \left\langle \nabla J^T\overrightarrow{N}, \frac{\frac{D\phi_S}{D\vec{b}}\cdot\frac{d\vec{b}}{dt}}{\langle\overrightarrow{N},\nabla_{\vec{z}}\phi_S\rangle}\right\rangle_p \\
&= -\left\langle \nabla J^T\overrightarrow{N}, \frac{\frac{D\phi_S}{D\mathbf{A}^{-1}}}{\langle\overrightarrow{N},\nabla_{\vec{z}}\phi_S\rangle}\right\rangle_p \cdot\frac{d\mathbf{A}^{-1}}{dt} - \left\langle \nabla J^T\overrightarrow{N}, \frac{\frac{D\phi_S}{D\vec{b}}}{\langle\overrightarrow{N},\nabla_{\vec{z}}\phi_S\rangle}\right\rangle_p \cdot\frac{d\vec{b}}{dt} \\
&= \nabla_{\mathbf{A}^{-1}}J\cdot\frac{d\mathbf{A}^{-1}}{dt}+\nabla_{\vec{b}}J\cdot\frac{d\vec{b}}{dt}, \qquad \text{(D-6)}
\end{aligned}
$$

where

$$\nabla_{\mathbf{A}^{-1}}J = -\left\langle \nabla J^T\overrightarrow{N}, \frac{\frac{D\phi_S}{D\mathbf{A}^{-1}}}{\langle\overrightarrow{N},\nabla_{\vec{z}}\phi_S\rangle}\right\rangle_p, \qquad \text{(D-7)}$$

$$\nabla_{\vec{b}}J = -\left\langle \nabla J^T\overrightarrow{N}, \frac{\frac{D\phi_S}{D\vec{b}}}{\langle\overrightarrow{N},\nabla_{\vec{z}}\phi_S\rangle}\right\rangle_p, \qquad \text{(D-8)}$$

which gives Eqs. (16) and (17). In the gradient descent process, we can set $\frac{d\mathbf{A}^{-1}}{dt} = -\nabla_{\mathbf{A}^{-1}}J$ and $\frac{d\vec{b}}{dt} = -\nabla_{\vec{b}}J$.

References

[1] Darshan Bryner, Anuj Srivastava, Bayesian active contours with affine-invariant, elastic shape prior, in: The IEEE Conference on Computer Vision and Pattern Recognition, 2014.

[2] Darshan Bryner, Anuj Srivastava, Eric Klassen, Affine-invariant, elastic shape analysis of planar contours, in: IEEE Conference on Computer Vision and Pattern Recognition, 2012.

[3] V. Caselles, R. Kimmel, G. Sapiro, Geodesic active contour, Int. J. Comput. Vision 22 (1) (1997) 61–79.

[4] Vicent Caselles, Francine Catté, Tomeu Coll1, Françoise Dibos, A geometric model for active contours in image processing, Numer. Math. 66 (1) (1993) 1–31.

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

[6] Yunmei Chen, Hemant D. Tagare, Sheshadri Thiruvenkadam, Feng Huang, David Wilson, Kaundinya S. Gopinath, Richard W. Briggs, Edward A. Geiser, Using prior shapes in geometric active contours in a variational framework, Int. J. Comput. Vision 50 (3) (2002) 315–328.

[7] D. Cremers, F.R. Schmidt, F. Barthel, Shape priors in variational image segmentation: convexity, Lipschitz continuity and globally optimal solutions, in: IEEE Conference on Computer Vision and Pattern Recognition, 2008.

[8] Daniel Cremers, Timo Kohlberger, Christoph Schnörr, Shape statistics in kernel space for variational image segmentation, Pattern Recognit. 36 (9) (2003) 1929–1943.

[9] Navneet Dalal, Bill Triggs, Histograms of oriented gradients for human detection, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005.

[10] M. Dantone, J. Gall, G. Fanelli, L. Van Gool, Real-time facial feature detection using conditional regression forests, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2012.

[11] P. Etyngier, F. Segonne, R. Keriven, Shape priors using manifold learning techniques, in: Proceedings of the Eleventh IEEE International Conference on Computer Vision, 2007.

[12] Alban Foulonneau, Pierre Charbonnier, Fabrice Heitz, Affine-invariant geometric shape priors for region-based active contours, IEEE Trans. Pattern Anal. Mach. Intell. 28 (8) (2006) 1352–1357.

[13] Hon Pong Ho, Yunmei Chen, Huafeng Liu, Pengcheng Shi, Level set active contours on unstructured point cloud, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005.

[14] Hao Jiang, Mark S. Drew, Ze-Nian Li, Matching by linear programming and successive convexification, IEEE Trans. Pattern Anal. Mach. Intell. 29 (2007) 959–975.

[15] Hao Jiang, Stella X. Yu, David R. Martin, Linear scale and rotation invariant matching, IEEE Trans. Pattern Anal. Mach. Intell. 33 (7) (2011) 1339–1355.

[16] Shantanu H. Joshi, Anuj Srivastava, Intrinsic bayesian active contours for extraction of object boundaries in images, Int. J. Comput. Vision 81 (3) (2009) 331–355.

[17] A. Joulin, F. Bach, J. Ponce, Multi-class cosegmentation, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2012.

[18] Yan Ke, Rahul Sukthankar, PCA-SIFT: a more distinctive representation for local image descriptors, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004.

[19] Gunhee Kim, Eric P. Xing, Li Fei-Fei, Takeo Kanade,. Distributed cosegmentation via submodular optimization on anisotropic diffusion, in: Proceedings of the International Conference on Computer Vision, 2011.

[20] S. Lankton, A. Tannenbaum, Localizing region-based active contours, IEEE Trans. Image Process. 17 (11) (2008) 2029–2039.

[21] E.L. Lawler, Optimal cycles in doubly weighted linear graphs, in: Theory of Graphs: International Symposium, 1966, pp. 209–213.

[22] M.E. Leventon, W.E.L. Grimson, O. Faugeras, Statistical shape influence in geodesic active contours, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2000.

[23] Bing Li, Scott T. Acton, Automatic active model initialization via poisson inverse gradient, IEEE Trans. Image Process. 17 (8) (2008) 1406–1420.

[24] C. Li, C. Kao, J.C. Gore, Z. Ding, Minimization of region-scalable fitting energy for image segmentation, IEEE Trans. Image Process. 17 (10) (2008) 1940–1949.

[25] Chunming Li, Jundong Liu, Martin D. Fox, Segmentation of external force field for automatic initialization and splitting of snakes, Pattern Recognit. 38 (11) (2005) 1947–1960.

[26] Hongsheng Li, Edward Kim, Xiaolei Huang, Lei He, Object matching with a locally affine-invariant constraint, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010.

[27] David G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vision 60 (2004) 91–110.

[28] David G. Luenberger, Yinyu Ye, Linear and Nonlinear Programming, Second Edition, third ed., Springer, 2008.

[29] B.S. Morse, Weiming Liu, T.S. Yoo, K. Subramanian, Active contours using a constraint-based implicit representation, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, June 2005.

[30] Benoit Mory, Roberto Ardon, Anthony J. Yezzi, J. Thiran, Non-euclidean image-adaptive radial basis functions for 3d interactive segmentation, in: IEEE 12th International Conference on Computer Vision, 2009.

[31] Stanley Osher, James A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations, J. Comput. Phys. 79 (1988) 12–49.

[32] N. Paragios, R. Deriche, Geodesic active regions and level set methods for supervised texture segmentation, Int. J. Comput. Vision 46 (3) (2002) 223–247.

[33] N. Paragios, O. Mellina-Gottardo, V. Ramesh, Gradient vector flow fast geometric active contours, IEEE Trans. Pattern Anal. Mach. Intell. 26 (3) (2004) 402–407.

[34] Nikos Paragios, User-aided boundary delineation through the propagation of implicit representations, in: Medical Image Computing and Computer-Assisted Intervention (MICCAI), vol. 2879, 2003, pp. 678–686.

[35] Mikal Rousson, Nikos Paragios, Prior knowledge, level set representations & visual grouping, Int. J. Comput. Vision 76 (3) (2008) 231–243.

[36] Thomas Schoenemann, Daniel Cremers, Globally optimal image segmentation with an elastic shape prior, in: Proceedings of the 12th International Conference on Computer Vision, 2007.

[37] Thomas Schoenemann, Daniel Cremers, A combinatorial solution for model-based image segmentation and real-time tracking, IEEE Trans. Pattern Anal. Mach. Intell. 32 (2010) 1153–1164.

[38] Greg Slabaugh, Quynh Dinh, Gozde Unal, A variational approach to the evolution of radial basis functions for image segmentation, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.

[39] Paul Viola, Michael J. Jones, Robust real-time face detection, Int. J. Comput. Vision 57 (2004) 137–154.

[40] Junyan Wang, Kap Luk Chan, Shape evolution for rigid and nonrigid shape registration and recovery, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2009.

[41] Junyan Wang, Kap Luk Chan, Yadong Wang, On the stationary solution of PDE based curve evolution, in: Proceedings of the 19th British Machine Vision Conference, 2008.

[42] Xianghua Xie, M. Mirmehdi, Mac: magnetostatic active contour model, IEEE Trans. Pattern Anal. Mach. Intell. 30 (4) (2008) 632–646.

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

[44] Sai Kit Yeung, Pengcheng Shi, Stochastic inverse consistency in medical image registration, in: Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2005, pp. 188–196.

[45] Sai Kit Yeung, Chi-Keung Tang, Pengcheng Shi, Josien P.W. Pluim, Max A. Viergever, Albert C.S. Chung, Helen C. Shen, Enforcing stochastic inverse consistency in non-rigid image registration and matching, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2008.