

Coupled Parametric Active Contours

Christophe Zimmer, *Member, IEEE*, and
J.-C. Olivo-Marin, *Senior Member, IEEE*

Abstract—We propose an extension of parametric active contours designed to track nonoccluding objects transiently touching each other, a task where both parametric and single level set-based methods usually fail. Our technique minimizes a cost functional that depends on all contours simultaneously and includes a penalty for contour overlaps. This scheme allows us to take advantage of known constraints on object topology, namely, that objects cannot merge. The coupled contours preserve the identity of previously isolated objects during and after a contact event, thus allowing segmentation and tracking to proceed as desired.

Index Terms—Segmentation, tracking, active contours, topology, 2D video.

1 INTRODUCTION

ACTIVE contours provide a very effective and versatile framework for image segmentation and object tracking [1], [2], [3], [4]. These methods are based on the minimization of a cost functional, often called energy, that contains data attachment terms related to the position of the contour with respect to image features and regularizing terms, such as penalties on the contour length or area. This minimization proceeds iteratively, by letting the contour move away from an initial guess according to an evolution equation derived from the energy functional. The equilibrium position of the contour then provides a segmentation of the image which can be used as initial guess for the segmentation of the following frame when tracking objects in image sequences.

Two main types of active contours exist according to whether the contours are represented explicitly or implicitly. In the explicit representation used by the original active contours (also called snakes) [1], contours are parametric curves $C(p) = (x(p), y(p))$, i.e., functions from a scalar interval $[a, b] \subset \mathbb{R}$ into the image domain $\Omega \subset \mathbb{R}^2$ [1], [2], [3], [5], [6]. In the more recently popularized implicit approach, contours are obtained as the zero level set $\phi^{-1}(0) = \{(x, y) | \phi(x, y) = 0\}$ of a scalar function $\phi(x, y)$ defined over Ω [7], [8], [4], [9]. These two types of active contours differ considerably in their ability to handle object topology or multiple objects, computational efficiency, ease of implementation, and user interaction, as will be discussed in Section 2 below.

The applicative goal that motivated the present work is to track mobile biological cells from videomicroscopy image sequences in order to quantify their motion under varying experimental conditions (mutants, drugs, chemotactic gradients, etc.). This task requires both computational efficiency and the ability to control object topology. Computational efficiency is needed because of the large quantity of data to be analyzed (typically, several hundreds to thousands of images per experiment with many experiments needed to assess the effect of one condition or molecule). A control of object topology is needed because moving cells frequently touch each other in typical experiments. Since the intensities and textures of touching cells are often (though not necessarily) similar, it is useful, and often indispensable, to use the prior knowledge that

cells do not actually merge in order to obtain an acceptable segmentation. Without this external constraint, most segmentation methods will also merge the associated objects, even if the cells were spatially isolated and segmented as distinct objects earlier in the sequence. As will be argued in Section 2 below, existing parametric or implicit active contour methods cannot address all of these requirements simultaneously, mainly because level set methods are slow, while existing parametric active contours generally fail on touching objects.

Here, we present an extension of parametric active contours that combines the ability to handle transiently touching objects and exert topological control with computational efficiency and simplicity. Section 2 discusses related work; Section 3 describes a parametric active contour model for single (uncoupled) contours adapted from [1] and [4]. In Section 4, we extend this model to multiple coupled contours. Section 5 shows results on some example data and Section 6 concludes the paper.

2 RELATED METHODS

In this section, we discuss how existing active contours methods fare with respect to the two applicative requirements mentioned above, namely, computational speed and the ability to handle multiple objects undergoing frequent contacts.

A prominent difference between explicit and implicit active contours is the topological flexibility of level sets that stands in contrast with the topological rigidity of parametric contours. A standard parametric active contour [1], [2], [3], [10], [5], [6] consists of a single (usually closed) connected curve, which cannot readily split or merge with other contours unless relatively intricate “surgical” procedures are used (e.g., [11]). If several parametric contours are initialized in an image, these curves may evolve independently and cross each other, leading to multiply labeled regions and incorrect segmentations [12]. In contrast, the level set approach [7], [8], [13], [4], [9] can represent an arbitrary number of disjoint contours by a single function ϕ . The contour evolution equation is replaced by an evolution equation for ϕ . As ϕ evolves, the contours $\phi^{-1}(0)$ can split and merge automatically, while intersecting contours and multiple labels are ruled out by construction. Level set methods are often considered superior to parametric active contours, to a large extent because of this topological flexibility. Indeed, since an arbitrary initial contour (or a set of contours) can, in principle, turn into any number of object contours, initialization can be entirely automatic. For the same reason, objects not present on the first frame of a sequence can be detected at later stages without user assisted reinitialization. Finally, the method naturally adapts to objects that change topology over time, such as dividing cells or merging vesicles.

However, this flexibility can turn into a weakness in situations where a priori information on topology is needed to overcome ambiguities in interpreting the image. This was recently recognized [14] in the context of static brain image segmentation, where standard, topologically unconstrained level set techniques tend to produce spurious connected components at object boundaries where image information is weak. To produce segmentations consistent with a previously known topology, a topology preserving extension of the level set methodology was proposed and shown to improve the segmentation of either single objects such as cortical white matter or separate objects of similar image characteristics such as touching bone cells [14].

Another limitation concerns all implicit methods that employ only one level set function ϕ [8], [4], [14]. As we pointed out in [15], these methods are suboptimal for segmenting objects with distinct image characteristics, for instance, fluorescent cells with different image levels, because the image-dependent energy terms cannot

• The authors are with the Quantitative Image Analysis Group, Institut Pasteur, 25 rue du Docteur Roux, 75015 Paris, France.
E-mail: {czimmer, jcolivo}@pasteur.fr.

Manuscript received 4 Aug. 2004; revised 5 Mar. 2005; accepted 9 Mar. 2005; published online 14 Sept. 2005.

Recommended for acceptance by P. Torr.

For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0405-0804.

be tuned to individual objects (unless the contours are extracted at each evolution time step to allow identification of connected components, as in [16]). This is in contrast with parametric contours, which can be used straightforwardly to minimize independently defined energy functionals specific to individual objects. This limitation can also be overcome within the implicit framework by using multiple level set functions $\phi_i, i = 1 \dots N$ [17], [18], [19], [20], [15]. Multiple level set functions allow improved segmentation results by taking into account specific characteristics of individual objects [19], [15] or object classes [17], [18], [20]. However, if the different ϕ_i evolve independently of each other, their zero level sets are likely to intersect and create multiply labeled regions as in the case of parametric active contours. This is clearly undesired when segmenting nonoccluding objects. Vese and Chan [18] have proposed a method using vector-valued labels which allows N phases to be represented efficiently by only $\log_2(N)$ level set functions and guarantees a perfect partition of the image. However, these phases correspond to classes of similar intensities or textures, thus, objects sharing similar image characteristics (e.g., cells with similar levels of fluorescence) are treated as a single phase and topological constraints between them cannot be enforced. Note that the same limitation concerns other labeling techniques quite generally, even those in which the number of classes is fixed by the user.

Most other multiple level set techniques [17], [19], [15] represent N classes by N level set functions and introduce additional constraints in order to ensure, at least approximately, a partition of the image into nonoverlapping domains. These coupled level set methods are appropriate to address the difficulty posed by touching nonoccluding objects, whether these have similar image characteristics or not [15]. However, the use of multiple functions ϕ_i further exacerbates the already large computational complexity inherent to level set techniques. The most basic implementations require updating ϕ over the whole image domain at every time step of the discretized evolution equations. This is a huge computational disadvantage compared to the parametric methods, which mostly require updating the position of the contour control points only. Several numerical schemes have been proposed to reduce the computation time of level set methods, including narrow band and fast marching [21], [22] or additive operator splitting [23]. Despite substantial improvements in efficiency, however, implicit active contours are generally still many times if not orders of magnitude slower than corresponding parametric active contours (see, e.g., [11]).

In addition, the explicit representation makes parametric contours very amenable to user interaction, which is very helpful to correct or guide the segmentation process when needed [1]. This is less readily done with implicit methods since the contours must first be extracted from the level set function. Finally, level set methods require substantially more complex implementations for 2D images. These reasons justify the use of parametric active contours, especially in applications where very large 2D image sequences must be processed in reasonable time.¹ In the following, we therefore propose a scheme that couples multiple parametric active contours, thus combining the good segmentation properties of the coupled level sets techniques with the simplicity and efficiency of parametric methods.

3 UNCOUPLED PARAMETRIC ACTIVE CONTOURS

In this section, we first consider single (uncoupled) parametric active contours. Coupling will then be introduced in Section 4.

1. For 3D images, we prefer level set-based methods, as described in [26].

3.1 Gradient-Based Parametric Active Contours

We first briefly recall the original active contour model as introduced by [1]. It is based on the following energy functional:

$$E_0(\mathbf{C}) = \int_0^1 \left[\frac{1}{2} \alpha \left(\frac{\partial \mathbf{C}}{\partial p} \right)^2 + \frac{1}{2} \beta \left(\frac{\partial^2 \mathbf{C}}{\partial p^2} \right)^2 - f(\mathbf{C}(p)) \right] dp, \quad (1)$$

where α and β are constant positive parameters and f is an edge map of the image, such as the gradient magnitude of a smoothed version of the image intensity I . The last term in (1) is the data attachment term, whose minimization tends to push the contour toward pixels of high image gradient. The first two terms in (1) are regularization terms, whose minimization makes the contour act as a membrane that resists stretching (α term) and a thin plate that resists bending (β term) [1]. The minimization of (1) is achieved by letting the contour evolve from an initialization $\mathbf{C}_0 = \mathbf{C}(t=0)$ according to the evolution equation [1]:

$$\frac{\partial \mathbf{C}}{\partial t} = \alpha \frac{\partial^2 \mathbf{C}}{\partial p^2} - \beta \frac{\partial^4 \mathbf{C}}{\partial p^4} + \nabla f. \quad (2)$$

When the contour attains a steady state ($\frac{\partial \mathbf{C}}{\partial t} = 0$), it satisfies the Euler-Lagrange equation associated to the minimization of (1) and achieves a segmentation of the image.

3.2 Region-Based Parametric Active Contours

Despite its success in many applications, the original snake [1] suffers from well-known limitations, which include the need to initialize the contour close to the actual boundary of the object and the sensitivity of the segmentation to spurious edges generated by noise. These two shortcomings stem partly from the fact that the data attachment term in (1) only depends on the image gradient along the contour and is thus “blind” to the rest of the image. Sensitivity to initialization and noise can be greatly reduced by using a data attachment term that depends on image quantities integrated over the whole image domain Ω , rather than on quantities integrated only along the contour length. A particularly appealing method of this type is the model of “active contours without edges” proposed by Chan and Vese [4], based on the well-known (piece-wise constant) Mumford-Shah functional [24] for image segmentation.

In order to make the active contours robust to noise and initialization, we therefore replace the gradient based data attachment term of (1) by the two-region-based data attachment terms of [4]. The energy functional then becomes:

$$E_1(\mathbf{C}, c_{\text{in}}, c_{\text{out}}) = \frac{1}{2} \int_0^1 \left[\alpha \left(\frac{\partial \mathbf{C}}{\partial p} \right)^2 + \beta \left(\frac{\partial^2 \mathbf{C}}{\partial p^2} \right)^2 \right] dp + \lambda_{\text{in}} \int_{\text{inside}(\mathbf{C})} (I - c_{\text{in}})^2 d\sigma + \lambda_{\text{out}} \int_{\text{outside}(\mathbf{C})} (I - c_{\text{out}})^2 d\sigma, \quad (3)$$

where λ_{in} and λ_{out} are constant positive parameters, I is the local image intensity, $d\sigma$ is the elementary surface, and $c_{\text{in}}, c_{\text{out}}$ are unknown scalars. Using the Euler-Lagrange equations associated to the minimization of E_1 , the following evolution equation can be obtained:

$$\frac{\partial \mathbf{C}}{\partial t} = \alpha \frac{\partial^2 \mathbf{C}}{\partial p^2} - \beta \frac{\partial^4 \mathbf{C}}{\partial p^4} - \left[\lambda_{\text{in}}(I - c_{\text{in}})^2 - \lambda_{\text{out}}(I - c_{\text{out}})^2 \right] \frac{\partial \mathbf{C}}{\partial p} \mathbf{n}, \quad (4)$$

where \mathbf{n} is the normal unit vector of \mathbf{C} pointing outward, assuming that \mathbf{C} is parameterized counter-clockwise. The variables c_{in} and

c_{out} are given by the average intensity inside and outside, respectively, of the current contour \mathbf{C} :

$$c_{\text{in}}(t) = \langle I \rangle_{\text{inside}(\mathbf{C})}; \quad c_{\text{out}}(t) = \langle I \rangle_{\text{outside}(\mathbf{C})}, \quad (5)$$

where we use the notation $\langle I \rangle_A = (\int_A I d\sigma) / (\int_A d\sigma)$ for averages of the intensity over a region $A \subset \Omega$.

Thanks to the region-based data attachment terms in (3) and (5), this model is able to segment objects with ill-defined boundaries, such as fluorescent cells in noisy microscopy images, requiring only a difference in mean intensities between the objects and the background (see Section 5). A welcome property of the model is that these mean intensities need not be specified a priori, but are self-consistently computed (5) from the minimization of (3).

The surface integrals in (5) apparently imply a significant computational cost compared to the gradient oriented active contours of Section 3.1, making the efficiency advantage over implicit methods less obvious. In practice, however, this additional cost need not be paid. First, although the optimal values for c_{in} and c_{out} are unknown at the onset, their final values after segmentation of the first frame provide an excellent guess for the following frames because object and background intensities change little across consecutive frames. These coefficients can thus be held constant during processing of all but the first frame or be updated only after contour convergence, once per frame, to accommodate slow changes in background or object intensity. Second, one can replace Ω by a smaller domain $\omega \subset \Omega$ around the initial contour $\mathbf{C}(t=0)$ that must only be large enough to contain the object entirely. This further reduces the computation time of surface integrals and has the additional welcome consequence that c_{out} better reflects the average background intensity in the vicinity of the object—and, therefore, yields a better segmentation—if the background intensity varies across the image. Finally, note that the residual cost of processing the first frame could be further reduced by converting the region integrals into contour integrals using Green's theorem, as done, e.g., in [10], [6]. This way, only one region integration is required before segmentation rather than at each evolution time step.

4 COUPLED PARAMETRIC ACTIVE CONTOURS

Parametric active contour methods were originally used to segment and track single objects [1], [3], [2]. The most straightforward extension to N objects is to initialize N active contours $\mathbf{C}_1, \dots, \mathbf{C}_N$ on each object and let each \mathbf{C}_i minimize an energy functional that depends on \mathbf{C}_i only, e.g., by using (3) and (4) with $\mathbf{C} = \mathbf{C}_1, \dots, \mathbf{C} = \mathbf{C}_N$. However, as illustrated in Section 5, these independently evolving contours lead to undesired results when previously distinct objects touch each other. To prevent contours from crossing each other and absorbing touching objects, we define a scheme where contours are coupled through the use of a single energy function that depends on all N contours simultaneously:

$$\begin{aligned} E_2(\mathbf{C}_1, \dots, \mathbf{C}_N, c_{\text{in},1}, \dots, c_{\text{in},N}, c_{\text{out}}) = & \\ & \frac{1}{2} \sum_{i=1}^N \int_0^1 \left[\alpha \left(\frac{\partial \mathbf{C}_i}{\partial p} \right)^2 + \beta \left(\frac{\partial^2 \mathbf{C}_i}{\partial p^2} \right)^2 \right] dp \\ & + \lambda_{\text{in}} \sum_{i=1}^N \int_{\text{inside}(\mathbf{C}_i)} (I - c_{\text{in},i})^2 d\sigma \\ & + \lambda_{\text{out}} \int_{\text{outside}(\mathbf{C}_1) \cap \dots \cap \text{outside}(\mathbf{C}_N)} (I - c_{\text{out}})^2 d\sigma \\ & + \gamma \sum_{i=1}^N \sum_{j=i+1}^N \int_{\text{inside}(\mathbf{C}_i) \cap \text{inside}(\mathbf{C}_j)} d\sigma. \end{aligned} \quad (6)$$

For $\gamma = 0$, this functional is simply a generalization of (3) to multiple contours. Except for the term weighted by λ_{out} , it then coincides with the sum of (3) for all individual contours, $\sum_{i=1 \dots N} E_1(\mathbf{C}_i, c_{\text{in},i}, c_{\text{out}})$. For $N = 1$ and $\gamma = 0$, (6) reduces exactly to (3). An advantage of using multiple active contours to represent different objects, rather than a single level set function, is apparent from the fact that the values $c_{\text{in},1}, \dots, c_{\text{in},N}$ are not necessarily equal. As mentioned in Section 2 and [15], this allows better segmentation of objects that have different average intensity levels. The term weighted by λ_{out} accounts for the fact that there is a single background, which is now defined by the region exterior to all N contours. The new term, weighted by $\gamma > 0$, is equal to the summed areas of pair-wise intersecting contour interiors and, thus, penalizes contour overlaps. While this overlap penalty is similar to that of some of the multiple level set methods discussed in Section 2 [17], [19], [15], its incorporation into the framework of parametric active contours is new.

The N evolution equations associated to the minimization of (6) are as follows ($i = 1 \dots N$):

$$\begin{aligned} \frac{\partial \mathbf{C}_i}{\partial t} = & \alpha \frac{\partial^2 \mathbf{C}_i}{\partial p^2} - \beta \frac{\partial^4 \mathbf{C}_i}{\partial p^4} - \\ & \left[\lambda_{\text{in}} (I - c_{\text{in},i})^2 - \lambda_{\text{out}} (I - c_{\text{out}})^2 + \gamma \sum_{j=1 \dots N, j \neq i} \epsilon_j(\mathbf{C}_i) \right] \left| \frac{\partial \mathbf{C}_i}{\partial p} \right| \mathbf{n}_i, \end{aligned} \quad (7)$$

where \mathbf{n}_i is the outward normal to contour \mathbf{C}_i and $\epsilon_j(\mathbf{x})$ is the set indicator function of the interior of \mathbf{C}_j , that is, $\epsilon_j(\mathbf{x}) = 1$ if $\mathbf{x} \in \text{inside}(\mathbf{C}_j)$, otherwise $\epsilon_j(\mathbf{x}) = 0$. The coefficients $c_{\text{in},1}, \dots, c_{\text{in},N}, c_{\text{out}}$ evolve according to the following $N + 1$ equations:

$$\begin{aligned} c_{\text{in},i}(t) &= \langle I \rangle_{\text{inside}(\mathbf{C}_i)}, & i = 1 \dots N \\ c_{\text{out}}(t) &= \langle I \rangle_{\text{outside}(\mathbf{C}_1) \cap \dots \cap \text{outside}(\mathbf{C}_N)}. \end{aligned} \quad (8)$$

The interpretation of the new evolution equation (7) is straightforward when compared to its single contour equivalent (4): The term associated to the overlap penalty (γ term) is zero along portions of \mathbf{C}_i that are not enclosed by any of the other contours; along portions of \mathbf{C}_i located inside other contours, this term is a vector that points inwards, causing the contour to recede from regions occupied by others. The speed of this recession is proportional to the number of contours that overlap each other, thus, multiple overlaps will be reduced faster than simple overlaps.

The coupling between contours requires additional computations to evaluate ϵ_j . In our implementation, $\epsilon_j(\mathbf{C}_i(p))$ is determined simply by counting the intersections of a ray emanating from each control point $\mathbf{C}_i(p)$ with the polygons formed by the control points of the other contours $\mathbf{C}_j, j \neq i$ [25]. This counting is performed only for control points $\mathbf{C}_i(p)$ inside the bounding box of \mathbf{C}_j , otherwise, $\epsilon_j(\mathbf{C}_i(p))$ is set to zero—this leads to a dramatic reduction of the computational cost due to coupling. If needed, efficiency could be further improved by computing ϵ_j from binary masks associated to each contour, provided that these masks are updated only at pixels crossed by the contours during each evolution time step [10].

5 EXPERIMENTS

In this section, we illustrate the application of the coupled active contours on some examples. The first example (Fig. 1) shows the evolution of three coupled contours in the absence of image data and regularization, i.e., for $\alpha = \beta = \lambda_{\text{in}} = \lambda_{\text{out}} = 0, \gamma > 0$. In this case, the energy reduces to the overlap penalty. The contours do not move except where they overlap each other; in the overlapping regions, the contours recede from each other until they are disjoint, as expected.

Fig. 2 illustrates how coupled active contours improve the segmentation and tracking of two transiently touching objects. This

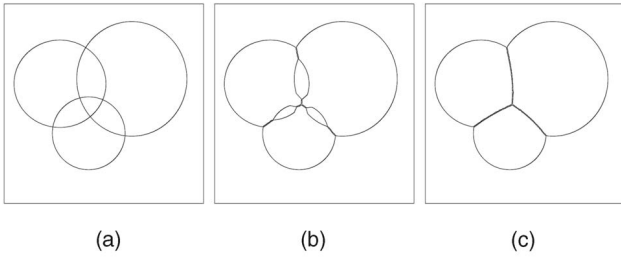


Fig. 1. Three contours evolving under the effect of the overlap penalty alone. (a) Initial contours. (b) Intermediate stage. (c) Final contours. Image size: 400×400 . Computation time: 0.6s.

example sequence shows two objects that are isolated on the first frame, touch each other in the next two frames, and are again isolated in the fourth frame. In this simple situation, two uncoupled parametric active contours lead to blatantly incorrect segmentations (Fig. 2, rows 2 and 3): As soon as the objects touch (time points t_2 and t_3), the two contours overlap and encompass both objects. This is expected, since the intensity difference between the two objects is smaller than the intensity difference between each object and the background. When the two objects move apart (t_4), each contour remains attached to both objects simultaneously. Even worse, at the gap between the objects, each contour intersects itself, fails to converge, and “blows apart,” as apparent from Fig. 2, rows 2 and 3 at time t_4 (only intermediate stages of contour evolution are shown for this time point). As a result, tracking breaks down completely. In contrast, coupling the

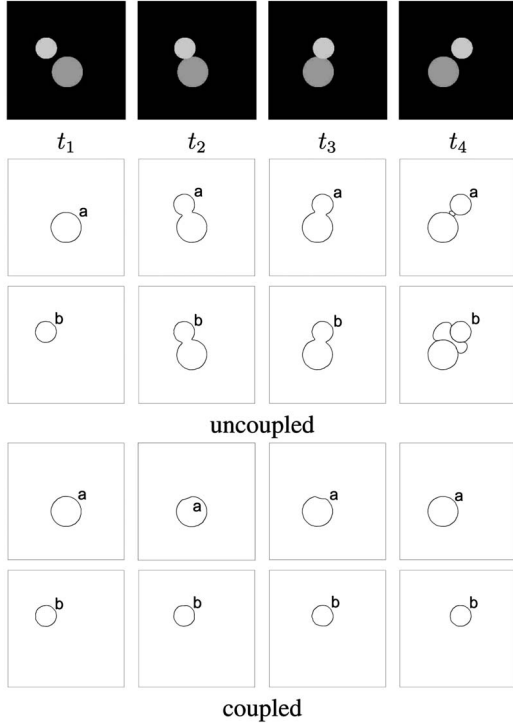


Fig. 2. Segmentation of two colliding objects by coupled versus uncoupled active contours. Row 1: Test image sequence (from left to right). Rows 2 and 3: Segmentation by two uncoupled active contours. Rows 4 and 5: Segmentation by two coupled active contours ($\gamma = 0.1$). Rows 2 and 4 show active contour a , initialized on the bottom object. Rows 3 and 5 show active contour b , initialized on the top object. Note how coupled active contours provide a satisfying segmentation, whereas uncoupled contours fail. (Intermediate stages of contour evolution are shown for t_4 in rows 2-3 because no convergence is attained.) Other parameters: $\alpha = 0$, $\beta = 0.1$, $\lambda_{in} = \lambda_{out} = 0.1$. Image size: 250×250 . Average computation time per frame: 0.2s.

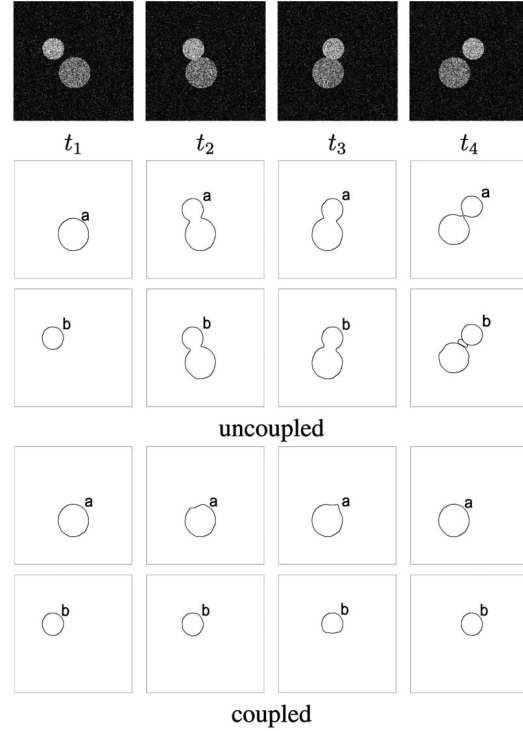


Fig. 3. Same as Fig. 2, but with Gaussian noise added to the images. Average computation time per frame: 0.4s.

active contours as described in Section 4 inhibits overlaps. As a result, each contour remains attached to its legitimate object, leading to much improved segmentations and tracking can proceed as desired (Fig. 2, rows 4 and 5).

Fig. 3 shows the same data corrupted by noise. Again, the uncoupled active contours fail (Fig. 3, rows 2-3). It is apparent (rows 4-5) that the coupled active contour method produces almost equally good results than on the noise-free data of Fig. 2. The only noticeable degradation is a slight misplacement of the boundary between the touching objects (compare Figs. 2 and 3 at time t_3 , rows 4-5, and see Section 6).

An example on real data is given in Fig. 4. The images are taken from a sequence obtained with a fluorescence microscope and show two migrating cells that are initially distant, then touch, and later move away from each other. Again, uncoupled contours produce incorrect segmentations as soon as the cells touch (Fig. 4, row 2, time t_2) and tracking subsequently breaks down (time t_3). With coupled active contours, however, a satisfying segmentation is obtained and cells are accurately tracked through the contact event (Fig. 4, rows 4-5). With our implementation and for the experiments reported here, processing times per frame (first frame excepted) range from fractions of a second to 1 second on a Sun Java workstation with two AMD Opteron 248 processors.

6 CONCLUSION AND FUTURE WORK

We have presented an extension of parametric active contours to track multiple objects that frequently touch each other as they move. Our method couples multiple active contours in a scheme that inhibits contour overlaps and maintains contours disjoint even though the tracked objects enter in contact. The coupling constraint is introduced directly as an overlap penalty into a single energy function dependent on all contours (in contrast to earlier and less general parametric coupling schemes [11], [12]). Although this can

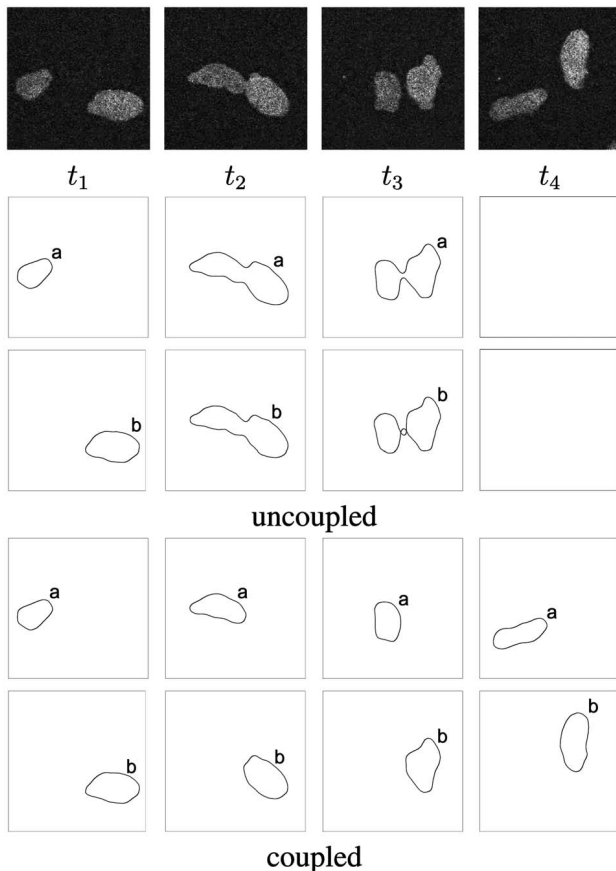


Fig. 4. Fluorescent cells in transient contact tracked by coupled (rows 4-5) versus uncoupled (rows 2-3) active contours. Intermediate contour evolution stages are shown in rows 2 and 3 at time t_3 . Parameters: $\alpha = 0$, $\beta = 0.1$, $\lambda_{in} = \lambda_{out} = 1$, $\gamma = \{0, 0.1\}$. Image size: 255×256 . Average computation time per frame: 0.9s.

also be achieved using multiple coupled level sets, the parametric approach is preferable for processing large image sequences thanks to its lower complexity. It is also more amenable to allowing user interaction and easier to implement for 2D data. A level set method such as [15] would, however, be useful as a complementary procedure to automatically initialize the contours and detect incoming objects at any time during the sequence.

Although our model maintains contours of touching objects disjoint, it does not guarantee that these contours accurately trace the interface between objects, as mentioned in the discussion of Fig. 3 above. In cases of very unfavorable initialization, it can even occur that one contour collapses while the other grows to enclose both objects. In practice, this extreme situation is easily prevented with the help of an additional penalty against excessive changes of the surface enclosed by each contour. Accurate localization of the interface for objects that can share very similar image characteristics, however, will require more specific data attachment terms or inclusion of stronger a priori constraints, for instance, on object shape (e.g., [5], [9]). Such constraints can be easily integrated with the coupled parametric active contours by adding new penalty terms to the cost functional (6). Thus, the presented method provides a basis for future work dedicated to segmentation and tracking of multiple nonoccluding objects undergoing frequent contacts. Although the primary applicative motivation of this work is biological cell imaging, the approach could be of interest to other applications where objects move in 2D images without overlapping each other, such as pedestrians or cars observed by a camera looking to the ground.

ACKNOWLEDGMENTS

The authors would like to thank the three anonymous referees for helpful comments, acknowledge useful discussions with Bo Zhang, and thank Nancy Guillén, Evelyne Coudrier, and François Amblard for the cell images in Fig. 4. This work was funded by Institut Pasteur.

REFERENCES

- [1] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active Contour Models," *Int'l J. Computer Vision*, vol. 1, pp. 321-331, 1988.
- [2] F. Leymarie and M.D. Levine, "Tracking Deformable Objects in the Plane Using an Active Contour Model," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 6, pp. 617-634, June 1993.
- [3] A. Blake and M. Isard, *Active Contours*. Springer, 1998.
- [4] T. Chan and L. Vese, "Active Contours without Edges," *IEEE Trans. Image Processing*, vol. 10, no. 2, pp. 266-277, 2001.
- [5] N. Ray, S. Acton, and K. Ley, "Tracking Leukocytes in Vivo with Shape and Size Constrained Active Contours," *IEEE Trans. Medical Imaging*, vol. 21, no. 10, pp. 1222-1235, Oct. 2002.
- [6] M. Jacob, T. Blu, and M. Unser, "Efficient Energies and Algorithms for Parametric Snakes," *IEEE Trans. Image Processing*, vol. 13, no. 9, pp. 1231-1244, Sept. 2004.
- [7] S. Osher and J. Sethian, "Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations," *J. Computational Physics*, vol. 79, pp. 12-49, 1988.
- [8] R. Malladi and J. Sethian, "A Unified Approach to Noise Removal, Image Enhancement, and Shape Recovery," *IEEE Trans. Image Processing*, vol. 11, no. 5, pp. 1554-1568, 1996.
- [9] N. Paragios, "A Level Set Approach for Shape-Driven Segmentation and Tracking of the Left Ventricle," *IEEE Trans. Medical Imaging*, vol. 22, no. 6, pp. 773-776, June 2003.
- [10] C. Chesnaud, P. Refregier, and V. Boulet, "Statistical Region Snake-Based Segmentation Adapted to Different Physical Noise Models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 11, pp. 1145-1157, Nov. 1999.
- [11] H. Delingette and J. Montagnat, "Shape and Topology Constraints on Parametric Active Contours," *Computer Vision and Image Understanding*, vol. 83, pp. 140-171, 2001.
- [12] C. Zimmer, E. Labruière, V. Meas-Yedig, N. Guillén, and J.-C. Olivo-Marin, "Segmentation and Tracking of Migrating Cells in Videomicroscopy with Parametric Active Contours," *IEEE Trans. Medical Imaging*, vol. 21, pp. 1212-1221, 2002.
- [13] N. Paragios and R. Deriche, "Geodesic Active Contours and Level Sets for the Detection and Tracking of Moving Objects," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 3, pp. 266-280, Mar. 2000.
- [14] X. Han, C. Xu, and J. Prince, "A Topology Preserving Level Set Method for Geometric Deformable Models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 6, pp. 755-768, June 2003.
- [15] B. Zhang, C. Zimmer, and J.-C. Olivo-Marin, "Tracking Fluorescent Cells with Coupled Geometric Active Contours," *Proc. IEEE Int'l Symp. Biomedical Imaging*, pp. 476-479, 2004.
- [16] B. Sebastian, H. Tek, J. Crisco, and B. Kimia, "Segmentation of Carpal Bones from CT Images Using Skeletally Coupled Deformable Models," *Medical Image Analysis*, vol. 7, no. 1, pp. 21-45, Mar. 2003.
- [17] C. Samson, L. Blanc-Féraud, G. Aubert, and J. Zerubia, "A Level Set Model for Image Classification," *Int'l J. Computer Vision*, vol. 40, no. 3, pp. 187-198, 2000.
- [18] L. Vese and T. Chan, "A Multiphase Level Set Framework for Image Segmentation Using the Mumford and Shah Model," *Int'l J. Computer Vision*, vol. 50, no. 3, pp. 271-293, 2002.
- [19] N. Paragios, "Coupled Geodesic Active Regions for Image Segmentation: A Level Set Approach," *European Conf. Computer Vision*, pp. 22-240, 2001.
- [20] A. Yezzi, A. Tsai, and A. Willsky, "A Fully Global Approach to Image Segmentation via Coupled Curve Evolution Equations," *J. Visual Comm. and Image Representation*, vol. 13, nos. 1/2, pp. 195-216, Mar./June 2002.
- [21] D. Adalsteinsson and J. Sethian, "A Fast Level Set Method for Propagating Interfaces," *J. Computational Physics*, vol. 118, no. 2, pp. 269-277, 1995.
- [22] J. Sethian, *Level Set Methods and Fast Marching Methods*. New York: Cambridge Univ. Press, 1999.
- [23] J. Weickert and G. Kuehne, "Fast Methods for Implicit Active Contour Models," *Geometric Level Set Methods in Imaging, Vision, and Graphics*, S. Osher and N. Paragios, eds., chapter 3, pp. 43-57, Springer, 2003.
- [24] D. Mumford and J. Shah, "Optimal Approximation by Piecewise Smooth Functions and Associated Variational Problems," *Comm. Pure and Applied Math.*, vol. 42, pp. 577-685, 1989.
- [25] J. O'Rourke, *Computational Geometry* in C. Cambridge Univ. Press, 1993.
- [26] A. Dufour, V. Shinin, S. Tajikbaksh, N. Guillén-Aghion, J.-C. Olivo-Marin, and C. Zimmer, "Segmenting and Tracking Fluorescent Cells in Dynamic 3-D Microscopy with Coupled Active Surfaces," *IEEE Trans. Image Processing*, vol. 14, no. 9, pp. 1396-1410, Sept. 2005.