HAL
open science

A robust parametric active contour based on Fourier descriptors

Tao Li, Alexandre Krupa, Christophe Collewet

▶ To cite this version:

A ROBUST PARAMETRIC ACTIVE CONTOUR BASED ON FOURIER DESCRIPTORS

Tao Li, Alexandre Krupa and Christophe Collewet

INRIA Rennes-Bretagne Atlantique, Campus de Beaulieu, 35042 Rennes cedex, France

ABSTRACT

This paper presents an optimal parametric active contour method based on Fourier descriptors to detect and track a contour from a sequence of images in real-time. The advantage of this approach is that not only convex but also concave contours can be considered. We also propose a method to initialize the active contour using only three user's clicks in the first image of the sequence and to automatically re-initialize it when possible topological changes are detected during the tracking. Moreover, the algorithm is implemented on GPU to ensure the real-time tracking performance.

Index Terms— Active contour, Fourier descriptors, contour tracking, ultrasound image

1. INTRODUCTION

An active contour, usually called snake, is an energy minimizing curve defined in image domain that evolves according to internal forces of the curve itself and external forces generated by the image. The snake was firstly introduced in [1] and has been widely applied in computer vision. In general, the snakes can be classified in two main categories: parametric snakes [2, 3] and geometric snakes [4, 5]. In our work we are interested in tracking the contour of an object of interest from a sequence of images in real-time. For that reason, we focus in this paper on parametric snakes instead of geometric ones which are not suitable for real-time application due to the high time-consuming process. In our previous work [6], such a parametric snake has been proposed. Indeed, by using a polar description we have shown that the snake evolution equations could be simplified to a diagonal system of ordinary differential equations. Unfortunately, in polar frame, only one point can be described by polar description for each angle. Thus the polar-snake can not describe concave contours, as for example the ones depicted in Fig.1. This can be done by B-spline parametric snakes as proposed in [3] for example, but its evolution equations system is no more a diagonal system. In this paper, we propose to model the active contour by using Fourier descriptors (FD) [7], which lead to a

diagonal system, and have the advantage to describe concave contours in a cartesian frame.

The main contribution of this paper is the determination of the evolution equations of the FD-based snake in order to detect and track an object of interest from an images sequence. Generally, a parametric active contour has to be initialized as nearest as possible to the real contour of the object of interest in order to converge to the correct position. Therefore, we propose as a second contribution of this paper, an approach to automatically initialize the snake. The method consists in initializing the snake to fit a set of contour points that are coarsely extracted using a fast Freeman chain coding algorithm (FCC) [8] applied on a binary image. After this automatic initialization, the FD-snake can finely evolve to reach the exact contour from the grey-level image by minimizing its energy criteria. Moreover, the coarsely extracted contour is also used to detect possible brutal topological changes of the object contour shape that require a FD-snake re-initialization.

To take into account the constraint of real-time contour detection, we also propose in this paper a parallel implementation of the FD-snake on GPU using the CUDA parallel computing architecture [9] to reduce the processing time.

This paper is structured as follows. First, the description of the parametric snake from FD and its evolution equations are developed in Section 2. Then the method we propose to automatically initialize the snake and to detect the topological changes of the object shape is presented in Section 3. Section 4 provides the principle of the snake implementation on GPU. Finally, experimental results of the contour extraction are presented and discussed in Section 5.

2. EVOLUTION EQUATIONS OF THE FD-SNAKE

2.1. Generic equation of snake

Let us define a generic curve model C_q of the parametric snake in the image plane by:

$$C_{\mathbf{q}}(u) : \mathbf{x}(u) = \mathbf{x}_c + \sum_{l=0}^{n} q_l \mathbf{\Phi}_l(u) \tag{1}$$

$C_{\mathbf{q}}$ is a linear form depending on a n-dimensional parameters vector $\mathbf{q}$ and $\mathbf{x}_c$ is a point inside the area delimited by the contour; $u \in [0, 2\pi]$ and $\mathbf{\Phi}_l$ ($l = 0, \ldots, n$) are 2D vector

functions that can be substituted by different descriptor terms, such as polar description, B-spline and Fourier descriptors.

As we did in [6], the evolution of the curve is modeled according to the Lagrangian mechanics formalism by considering $\mathbf{q}$ as the generalized coordinates of the system. The Euler-Lagrange equation of each component q_i is defined by:

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_i}\right) - \frac{\partial U}{\partial q_i} = Q_i \tag{2}$$

where T is the kinetic energy and U is the potential energy of the active contour. Q_i describes the generalized forces associated to q_i:

$$Q_i = \int_{C_\mathbf{q}} \mathbf{f}^\top(u) \frac{\partial C_\mathbf{q}(u)}{\partial q_i} du \tag{3}$$

with $\mathbf{f}$ the total of the dissipative forces defined by $\mathbf{f} = \mathbf{f}_{vis} + \mathbf{f}_{im}$. The forces $\mathbf{f}_{vis}$ and $\mathbf{f}_{im}$ are respectively due to a viscous friction and the image itself. Here, we consider $\mathbf{f}_{im} = -\nabla E_{im}$ where E_{im} is the potential energy term in the image $\mathbf{I}$: $E_{im} = -\|\nabla \mathbf{I}\|$, with $\|\nabla \mathbf{I}\|$ the norm of the image gradient $\nabla \mathbf{I}$.

Finally, (2) can be written under a vectorial form (see [6] for more details):

$$\mathbf{M\ddot{q}} + \mathbf{C\dot{q}} + \mathbf{Kq} = \mathbf{Q}_{im}(\mathbf{q}) = -\int_{C_\mathbf{q}} \nabla E_{im}^\top(u) \frac{\partial C_\mathbf{q}(u)}{\partial \mathbf{q}} du \tag{4}$$

where $\mathbf{M} = [M_{mn}]$, $\mathbf{C} = [C_{mn}]$ and $\mathbf{K} = [K_{mn}]$ with:

$$\begin{cases} M_{mn} = \mu \int_{C_\mathbf{q}} \mathbf{\Phi}_m^\top \mathbf{\Phi}_n du \\[2mm] C_{mn} = \gamma \int_{C_\mathbf{q}} \mathbf{\Phi}_m^\top \mathbf{\Phi}_n du \\[2mm] K_{mn} = k_1 \int_{C_\mathbf{q}} \mathbf{\Phi}_m'^\top \mathbf{\Phi}_n' du + k_2 \int_{C_\mathbf{q}} \mathbf{\Phi}_m''^\top \mathbf{\Phi}_n'' du \end{cases} \tag{5}$$

with the mass density μ, the viscosity of the medium γ. The scalar k_1 tends to limit the extension of $C_\mathbf{q}$ while k_2 tends to limit its curvature. Replacing $\mathbf{\Phi}_l(u)$ of (1) by orthogonal functions whose first and second derivatives are also orthogonal can simplify (5) to diagonal matrices.

As proposed in [2], we set the mass density μ to zero to cancel the inertial behavior of the active contour in order to improve its dynamic stability. The differential system (4) can then be solved as a discrete optimization problem using explicit Euler's method leading to the following recursive expression of $\mathbf{q}$:

$$\mathbf{q}_{k+1} = \mathbf{q}_k + \Delta t \mathbf{C}^{-1}(\mathbf{Q}_{im}(\mathbf{q}_k) - \mathbf{Kq}_k) \tag{6}$$

where $\mathbf{q}_{k+1}$ is the new estimate of the generalized coordinates, $\mathbf{q}_k$ is the previous estimate and Δt is the sampling rate.

2.2. Parametrization using the Fourier descriptors

Replacing the term $\mathbf{\Phi}_l$ of (1) by Fourier descriptors [7], the curve model C_q becomes:

$$C_\mathbf{q}(u) : \mathbf{x}(u) = \begin{pmatrix} a_0 \\ c_0 \end{pmatrix} + \sum_{k=1}^{h} \begin{pmatrix} a_k & b_k \\ c_k & d_k \end{pmatrix} \begin{pmatrix} \cos ku \\ \sin ku \end{pmatrix} \tag{7}$$

where h is the number of harmonics. By developing (7), we have $\mathbf{q} = (a_0, a_1 \cdots a_h, b_1 \cdots b_h, c_0, c_1 \cdots c_h, d_1 \cdots d_h)$ and consequently:

$$\begin{cases} 0 \leq l < h : \mathbf{\Phi}_l(u) = \begin{pmatrix} \cos lu \\ 0 \end{pmatrix} \\[3mm] h+1 \leq l < 2h : \mathbf{\Phi}_l(u) = \begin{pmatrix} \sin(l-h)u \\ 0 \end{pmatrix} \\[3mm] 2h+1 \leq l < 3h+1 : \mathbf{\Phi}_l(u) = \begin{pmatrix} 0 \\ \cos(l-2h-1)u \end{pmatrix} \\[3mm] 3h+2 \leq l < 4h+2 : \mathbf{\Phi}_l(u) = \begin{pmatrix} 0 \\ \sin(l-3h-1)u \end{pmatrix} \end{cases} \tag{8}$$

Furthermore, by substituting (8) in (5), the matrices $\mathbf{C}$ and $\mathbf{K} = \mathbf{K}_1 + \mathbf{K}_2$ involved in (4) become diagonal and constant:

$$\begin{cases} \mathbf{C} = \gamma \, \mathtt{diag}(2\pi, \pi \ldots \pi, \pi \ldots \pi, \\ \qquad\qquad 2\pi, \pi \ldots \pi, \pi \ldots \pi) \\[3mm] \mathbf{K}_1 = k_1 \, \mathtt{diag}(0, \pi \ldots \pi h^2, \pi \ldots \pi h^2, \\ \qquad\qquad 0, \pi \ldots \pi h^2, \pi \ldots \pi h^2) \\[3mm] \mathbf{K}_2 = k_2 \, \mathtt{diag}(0, \pi \ldots \pi h^4, \pi \ldots \pi h^4, \\ \qquad\qquad 0, \pi \ldots \pi h^4, \pi \ldots \pi h^4) \end{cases} \tag{9}$$

For that reason, the evolution equations system (4) becomes optimal and the computation time of (6) highly decreases. In addition, to ensure a convergence of the snake toward the true contour, even in the case of an initialization far away from it, we propose to add a surface-based energy term as in [6]. It can be simply done since the surface $S_\mathbf{q}$ can be analytically computed. Indeed we have:

$$S_\mathbf{q} = \pi\left(\sum_{k=1}^{h} k(a_k d_k - b_k c_k)\right) \tag{10}$$

3. AUTOMATIC INITIALIZATION AND TOPOLOGICAL CHANGE DETECTION

As mentioned in the introduction, the initialization of active contour and possible topological changes of the curve during contour tracking are two important issues. To deal with them, we propose to automatically initialize the snake from a set of contour points that are roughly extracted.

This extraction is done by binarizing the current image using an optimal adaptive threshold that we compute thanks to the Otsu's method [10]. Only the grey-level pixels included in an image bounding box (Fig.1(a)) are taken into account for the threshold adaptation.

The coarse contour $L_p(t)$ and the surface $A(t)$ of the object of interest in the binary image are then obtained by applying the fast Freeman chain coding algorithm [8]. At time

$t = 0$, the snake is then initialized from $L_p(t = 0)$ in the first image. Furthermore, the snake will be automatically re-initialized if the object of interest separates in two parts (Fig.2) or a large difference between $A(t)$ and $A(t - \Delta t)$ is detected from the following images of the sequence (Fig.3).

4. IMPLANTATION ON GPU

In order to perform real time contour tracking from a sequence of ultrasound images, for example in the context of an image-guided robotic medical application [11], we propose in this section a GPU implementation solution of the snake.

In (4), the diagonal matrices $\mathbf{C}$ and $\mathbf{K}$ are constant. $\mathbf{q}_k$ is the optimized result obtained from previous iteration. So we are interested to perform parallel computation of the term $\mathbf{Q}_{im}(\mathbf{q}) = -\int_{C_q} \nabla \mathbf{E}_{im}^\top(u) \frac{\partial C_q(u)}{\partial \mathbf{q}} du$. In this paper we consider that $\nabla \mathbf{E}_{im} = -(\|\nabla \mathbf{I}\|_x, \|\nabla \mathbf{I}\|_y)$ leading to:

$$\mathbf{Q}_{im}(\mathbf{q}) = -\sum_{n=1}^{N} \frac{\partial x(n)}{\partial q_i} \|\nabla \mathbf{I}\|_x(n) - \sum_{n=1}^{N} \frac{\partial y(n)}{\partial q_i} \|\nabla \mathbf{I}\|_y(n) \quad (11)$$

with N the points number of the snake. The partial derivation terms $\frac{\partial x(n)}{\partial q_i}$ and $\frac{\partial y(n)}{\partial q_i}$ are constant according to FD.

To gain time-consuming performance, the following algorithm was implemented on GPU:

Step 1: Copy the constant terms $\mathbf{C}$, $\mathbf{K}$, $\frac{\partial x(n)}{\partial q_i}$ and $\frac{\partial y(n)}{\partial q_i}$ from host memory (CPU) to device memory (GPU).
Step 2: If $k = 0$, copy $\mathbf{q}_{k=0}$ from CPU to GPU.
Step 3: Copy the current image from CPU to GPU.
Step 4: Compute $\|\nabla \mathbf{I}\|$ in parallel, the image gradient norm of each pixel is computed by a different GPU thread, therefore all the pixels can be processed on GPU in parallel.
Step 5: Compute the spatial gradient ∇E_{im} in parallel.
Step 6: Compute $\mathbf{Q}_{im}$ in parallel.
Step 7: Compute $\mathbf{q}_{k+1}$ in parallel.
Step 8: If $|\mathbf{q}_{k+1} - \mathbf{q}_k| > \epsilon$, then $k = k+1$ and go to Step 5.
Step 9: If $|\mathbf{q}_{k+1} - \mathbf{q}_k| \leq \epsilon$, copy $\mathbf{q}_{k+1}$ from GPU to CPU.
Step 10: Go to step 3 for the next image of the sequence.

5. EXPERIMENTAL RESULTS

All the following experiments were performed with a computer equipped with a dual core CPU Intel Xeon 5160, 4Gb memory and a graphic card NVIDIA GeForce GTX 285 providing a 240 cores GPU.

Fig.1(a) shows that only 3 points are required to be defined by the user in the first image of the sequence to initialize the snake. The point P_1 indicates the object of interest, the points P_2 and P_3 define the position of the bounding box (the white rectangle in Fig.1.(a)) where the adaptive threshold will be computed. The red line around the joystick cover shows the snake result after convergence in the image. Fig.1(b) and Fig.1(c) are results of the polar-snake and the FD-snake with

an echographic concave object section. In Fig.1(b) we can see that the polar-snake diverges while the FD-snake perfectly adapts to the true contour of the concave object both for the example of Fig.1(a) and Fig.1(c).


(a)
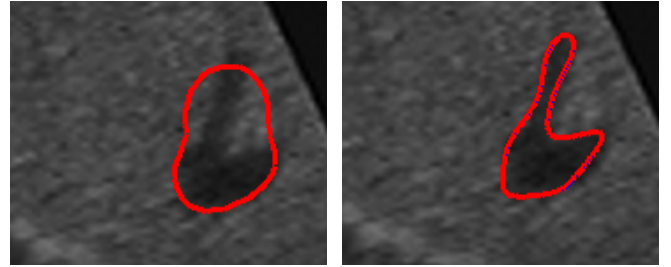
(b)   (c)

Fig. 1. (a): Initialization of the FD-snake from three clicks (with $h = 19$). (b) Polar-snake divergence on a concave US section (with $h = 6$). (c) FD-snake convergence on the same concave US section (with $h = 6$).

Fig.2 shows the results obtained from a sequence of US images where a fission of the object of interest was observed. Two coarsely extracted contours L_{G_1} and L_{G_2} are extracted from two different germs G_1 and G_2 used by the FCC. If the obtained contours L_{G_1} and L_{G_2} are different, we consider that the tracked object separates in two parts (Fig.2.(c)) and in this case we automatically initialize a second snake (the red one in Fig.2.(c)).

Fig.3 shows the results obtained from a sequence of US images where a topological change of the gallbladder was observed. In Fig.3(a), the gallbladder is seen as an ellipse at the beginning of the US images sequence. During the object tracking, we apply a rotation motion to the US probe in order to see the gallbladder as a tube (Fig.3(d)). Thanks to the method proposed in Section 3, the FD-snake can track this topological change by automatically re-initialize itself when a large difference of the coarse contour surface $A(t)$ extracted by the FCC is observed between 2 successive images.

Moreover we compare the time-consuming of the CPU
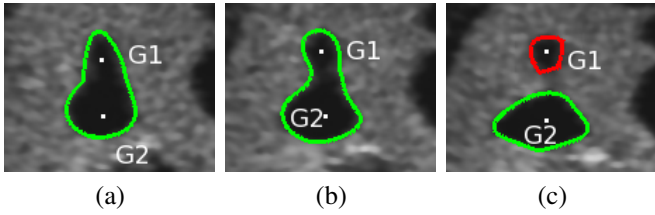
Fig. 2. Topological change (fission) detection of the object of interest and an automatically initialization of the second snake along a sequence of US images (with $h = 6$).
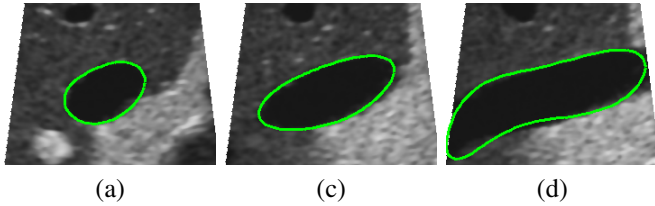
Fig. 3. Topological change detection of the object of interest and automatic re-initialization of the active contour during contour tracking along a sequence of US images (with $h = 6$).

and GPU algorithm of the FD-snake during the tracking of the gallbladder (Fig.3). Table1 reports the processing time of the CPU and GPU algorithm with different harmonics number h and points number N of the active contour. As shown in Table1, the GPU implementation consumes less time than the CPU for all experiments. Moreover, when h and N increase, the consuming time of the GPU implementation slightly increases but still ensures the real-time constraint. Note that this saving time can be used to increase h and / or N.

Consuming Time			
h	N	CPU algorithm	GPU algorithm
3	100	75ms	29ms
	400	82ms	35ms
6	100	80ms	31ms
	400	95ms	40ms

Table 1. The consuming time of CPU and GPU algorithm in function of harmonic number h and points number N.

6. CONCLUSION

We have presented in this paper an optimal FD-based parametric active contour which has the advantage to track both convex and concave contours. A method is also proposed to semi-automatically initialize the snake in the first image and automatically re-initialize it when topological changes are detected in the current image. Finally, we provided a GPU implementation solution to perform the contour tracking in real-time. In a future work we plan to develop a visual servoing method based on the FD-snake to control a medical robot equipped with an US probe (see [11]).

7. REFERENCES

[1] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[2] D. Terzopoulos and D. Metaxas, "Dynamic 3d models with local and global deformations: deformable superquadrics," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 13, no. 7, pp. 703 –714, 1991.

[3] F. Precioso, M. Barlaud, T. Blu, and M. Unser, "Robust real-time segmentation of images and videos using a smooth-spline snake-based algorithm," *IEEE Trans. on Image Processing*, vol. 14, no. 7, pp. 910 –924, July 2005.

[4] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *Int. Journal of Computer Vision*, vol. 22, no. 1, pp. 61–79, February 1997.

[5] C. Xu, Jr Yezzi, A., and J.L. Prince, "A summary of geometric level-set analogues for a general class of parametric active contour and surface models," in *IEEE Workshop on Variational and Level Set Methods in Computer Vision.*, 2001, pp. 104 –111.

[6] C. Collewet, "Polar snakes: A fast and robust parametric active contour model," in *IEEE Int. Conf. on Image Processing*, Cairo, Egypt, November 2009, pp. 3013–3016.

[7] S. Impedovo, B. Marangelli, and A. M. Fanelli, "A fourier descriptor set for recognizing nonstylized numerals," *IEEE Tran. on Systems, Man and Cybernetics*, vol. 8, no. 8, pp. 640 –645, August 1978.

[8] H. Freeman and L.S. Davis, "A corner-finding algorithm for chain-coded curves," *IEEE Trans. on computers*, vol. C-26, no. 3, pp. 297 –303, March 1977.

[9] Z. Yang, Y. Zhu, and Y. Pu, "Parallel image processing based on cuda," in *Int.Conf. on Computer Science and Software Engineering*, dec. 2008, vol. 3, pp. 198 –201.

[10] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 9, no. 1, pp. 62 –66, January 1979.

[11] R. Mebarki, A. Krupa, and F. Chaumette, "2d ultrasound probe complete guidance by visual servoing using image moments," *IEEE Trans. on Robotics*, vol. 26, no. 2, pp. 296–306, April 2010.