J Real-Time Image Proc (2018) 14:793–802
https://doi.org/10.1007/s11554-015-0531-5

ORIGINAL RESEARCH PAPER

Fast and accurate circle tracking using active contour models

Carmelo Cuenca[1] · Esther González[1] · Agustín Trujillo[1] · Julio Esclarín[1] · Luis Mazorra[1] · Luis Alvarez[1] · Juan Antonio Martínez-Mera[2] · Pablo G. Tahoces[2] · José M. Carreira[3]

Received: 31 December 2014 / Accepted: 10 September 2015 / Published online: 5 October 2015

Abstract In this paper, we deal with the problem of circle tracking across an image sequence. We propose an active contour model based on a new energy. The center and radius of the circle is optimized in each frame by looking for local minima of such energy. The energy estimation does not require edge extraction, it uses the image convolution with a Gaussian kernel and its gradient which is computed using a GPU–CUDA implementation. We propose a Newton–Raphson type algorithm to estimate a local minimum of the energy. The combination of an active contour model which does not require edge detection and a GPU–CUDA implementation provides a fast and accurate method for circle tracking. We present some experimental results on synthetic data, on real images, and on medical images in the context of aorta vessel segmentation in computed tomography (CT) images.

Keywords Circle · Tracking · Active Contour Models · snakes · GPU–CUDA

1 Introduction

Active contour model is a widely used technique for object detection and tracking. Given an initial object contour curve C, the active contour model optimizes the contour location by minimizing an energy. A general form of such energy is given by

$$E(C) = \oint_C g(\nabla I(C(s)), C'(s))\mathrm{d}s, \tag{1}$$

usually, function $g()$ is designed to attains its minima in high image gradient contour curves. In this paper, we assume that C is given by the contour boundary of a disk $D_R(c_x, c_y)$ and then the above energy minimization problem is strongly simplified and can be expressed as

Electronic supplementary material The online version of this article (doi:10.1007/s11554-015-0531-5) contains supplementary material, which is available to authorized users.

✉ Luis Alvarez
lalvarez@dis.ulpgc.es

Carmelo Cuenca
ccuenca@ctim.es

Esther González
egonzalez@ctim.es

Agustín Trujillo
atrujillo@ctim.es

Julio Esclarín
jesclarin@ctim.es

Luis Mazorra
lmazorra@ctim.es

Juan Antonio Martínez-Mera
juanantonio.martinez@usc.es

Pablo G. Tahoces
tahoces@usc.es

José M. Carreira
josemartin.carreira@usc.es

[1] Departamento de Informática y Sistemas, CTIM: Centro de tecnologías de la Imagen, Universidad de Las Palmas de Gran Canaria, Campus de Tafira, 35017 Las Pamas, Spain

[2] Centro Singular de Investigación en Tecnoloxías da Información (CITIUS), University of Santiago de Compostela, Santiago de Compostela, Spain

[3] University Hospital Complex of Santiago de Compostela (CHUS), Santiago de Compostela, Spain

$$E(R,\bar{c}) = \int_0^{2\pi} g(\nabla I(C_{R,\bar{c}}(\theta)), C'_{R,\bar{c}}(\theta))R\mathrm{d}\theta, \quad (2)$$

where $\bar{c} = (c_x, c_y)$ is the circle center and

$$C_{R,\bar{c}}(\theta) = (c_x + R \cdot cos(\theta), c_y + R \cdot sin(\theta)). \quad (3)$$

In general, active contour models are composed of two terms: an image fitting term which makes the curve evolve towards high contrast object boundaries and a regularization term which keeps the curve smooth. In our case, as we deal with circles, the regularization term is not required.

In this paper, we propose a new active contour model based on a new energy to track circles in an image sequence. The proposed energy, which is presented in detail in Sect. 3, includes a contour integral like the one presented in (2) as well as double integrals on contour neighborhood domains in both sides of the circle contour to measure the variation of the image in such domains. To estimate the local minima of the energy we propose a Newton–Raphson type minimization algorithm.

This paper is organized as follows: in Sect. 2, we present briefly some related works. In Sect. 3, the proposed active contour model is introduced. In Sect. 4, we present a description of the associated algorithm. In Sect. 5, we show some experimental results on synthetic and real images and finally, in Sect. 6, we present the main conclusions.

2 Related works

Active contour model is a widely used segmentation technique based on curve evolution, see for instance [3, 4, 6–8, 29, 32]. In the case of the curve we are interested in has a particular shape (polygons, circles, etc..) then we can use parametric active contour models where the model is simplified according to the curve parametrization, see for instance [5, 14]. Concerning the shape of function $g()$ in (1), several choices have been proposed in the literature. For instance, in [9], in the context of image segmentation, authors propose to use

$$g(\|\nabla I_\sigma\|), \quad (4)$$

where I_σ represents a convolution of the original image with a Gaussian kernel of standard deviation σ and $g()$ is a decreasing function. A typical choice of $g()$ commonly used in the literature is

$$g(s) = \frac{1}{1+\lambda s^2}, \quad (5)$$

where $\lambda \geq 0$. In [21] authors propose the following choice of $g()$ in (1) to define the active contour model

$$g(\nabla I(C(s)), C'(s)) = \nabla I(C(s)) \cdot \bar{n}(s), \quad (6)$$

where $\bar{n}(s)$ is the outward normal to the curve. Authors assume that the interior region of the curve is brighter than the exterior region of the curve. In that case, by minimizing the energy the curve moves towards locations with high gradient magnitude and with a gradient orientation similar to the curve normal. We point out that if $C(s)$ is clock-wise oriented with respect to s then

$$\bar{n}(s) = \frac{(C'(s)_y, -C'(s)_x)^T}{|C'(s)|}. \quad (7)$$

In [10], authors propose to use a region based fitting energy term given by

$$\alpha_- \int_{\mathrm{inside}(C)} (I(\bar{x}) - I_-)^2 + \alpha_+ \int_{\mathrm{outside}(C)} (I(\bar{x}) - I_+)^2, \quad (8)$$

where $inside(C)$ is the interior region of the curve, $outside(C)$ is the exterior region of the curve, I_- is the average of I in $inside(C)$ and I_+ is the average of I in $outside(C)$. In this case, by minimizing the energy, the curve tends to move towards locations where the image has a minimal variance in both sides of the curve. The curve location is optimal when the image is constant in both sides of the curve.

In [2], authors propose to use in the above energy a local neighborhood of the curve C as domain to evaluate the integral. That is, for instance, $inside(C)$ is replaced by

$$inside(C) \cap \{\bar{x} \in R^2 : \mathrm{d}(\bar{x}, C) < d_{max}\}. \quad (9)$$

The Hough Transform (HT) was first introduced in [18] as a method for detecting complex patterns of points in a digital image. Because it requires that the search patterns are described in a specific parametric form, it has been most commonly used for the detection of regular curves such as lines, circles, or regular shapes, see for instance [16, 18, 23, 31, 36]. HT have many desirable features, such as the possibility to recognize partial or slightly deformed shapes including missing parts of an object, the possibility to detect several instances of a particular shape occurring in different places of the image with independence of its size/orientation. However, HT has also some disadvantages such as the computing and storage requirements formerly commented or the detection of background structures from the boundaries of shapes other than those searched for [19] when signal to noise ratio of the image decreases [12]. In this sense, in [20] authors focus on dealing with this type of limitations, achieving results that improve those accomplished with the original algorithm. More recently, in [34], authors propose a method to extract line segments and elliptical arcs which is not based on the Hough transform. The method uses *a contrario* validation strategy which

formally guarantees the control of the number of false positives and requires no parameter tuning.

In this paper, we use a GPU-CUDA implementation strategy to speed up the proposed method. Nowadays, GPU implementation is a commonly used technique to parallelize and speed up computer vision algorithms. For instance, in [37] authors propose a GPU implementation of circle Hough transform, in [15, 17, 25–27] authors propose GPU implementation for object segmentation and detection and in [13, 22, 38] authors propose GPU-CUDA implementations for image denoising and restoration.

3 Active contour models for circle tracking

Let l_σ be the convolution of the original image I with a Gaussian kernel of standard deviation σ and $D_R(c_x, c_y)$ a disk of radius R and center $\bar{c} = (c_x, c_y)$. We propose to use the following energy to track circles in the image composed of a contour integral along the circle contour and 2 double integrals on domains in both sides of the circle contour.

$$E(R, c_x, c_y) = \tag{10}$$

$$\frac{1}{2\pi}\int_0^{2\pi} \nabla I_\sigma(C_{R,\bar{c}}(\theta)) \cdot \bar{n}(\theta)\mathrm{d}\theta \tag{11}$$

$$+\alpha_-\left(\frac{1}{A_-}\int_0^{2\pi}\int_{R-\delta_R^-}^{R}(I_\sigma(C_{r,\bar{c}}(\theta)) - I_-)^2 r\mathrm{d}r\mathrm{d}\theta\right)^{\frac{1}{2}} \tag{12}$$

$$+\alpha_+\left(\frac{1}{A_+}\int_0^{2\pi}\int_{R}^{R+\delta_R^+}(I_\sigma(C_{r,\bar{c}}(\theta)) - I_+)^2 r\mathrm{d}r\mathrm{d}\theta\right)^{\frac{1}{2}}, \tag{13}$$

where $\alpha_-, \alpha_+, \delta_R^- \geq 0$ and A_-, A^+ are the areas of the integral domains in both sides of the circle contour:

$$A_- = \pi\delta_R^-(2R - \delta_R^-), \qquad A_+ = \pi\delta_R^+(2R + \delta_R^+), \tag{14}$$

δ_R^+ is fixed in order $A_- = A_+$. Therefore,

$$\delta_R^+ = \sqrt{R^2 + 2\delta_R^- R - (\delta_R^-)^2} - R, \tag{15}$$

I_- and I_+ are the average of l_σ in the integral domains

$$I_- = \frac{1}{A_-}\int_0^{2\pi}\int_{R-\delta_R^-}^{R} I_\sigma(C_{r,\bar{c}}(\theta))r\mathrm{d}r\mathrm{d}\theta, \tag{16}$$

$$I_+ = \frac{1}{A_+}\int_0^{2\pi}\int_{R}^{R+\delta_R^+} I_\sigma(C_{r,\bar{c}}(\theta))r\mathrm{d}r\mathrm{d}\theta. \tag{17}$$

We point out that the term (11) is similar to the one proposed in [21], the main difference is that we normalize the energy in order it be independent of the boundary circle length and we introduce a Gaussian kernel convolution to smooth the image and also to help to the minimization scheme to find the proper local minimum. As in [21] we assume that the interior area of the circle we are interested in is brighter than the circle background. In the opposite case, we have to just change the sign of the integral to have the proper energy to minimize. Moreover, term (11) has the interesting property that, using the classical divergence theorem, it can be interpreted as

$$\frac{1}{2\pi}\int_0^{2\pi} \nabla I_\sigma(C_{R,\bar{c}}(\theta)) \cdot \bar{n}(\theta)d\theta = \frac{1}{2\pi R}\int_{D_R(c_x,c_y)} \Delta I_\sigma(x, y)\mathrm{d}x\mathrm{d}y.$$

Terms (12) and (13) of the energy are similar to the ones proposed in [10]. The main difference is that in each integral we introduce the power 1 / 2 on the integrals results. By introducing this change, we obtain that the global energy (10) is invariant under global contrast image variation in the sense that given an image I, and $\tilde{I} = \lambda I$ for any constant $\lambda > 0$, then the associated energies $E(R, c_x, c_y)$ and $\tilde{E}(R, c_x, c_y)$ given by (10) for both images satisfy:

$$\tilde{E}(R, c_x, c_y) = \lambda E(R, c_x, c_y), \tag{18}$$

in particular, the locations where $\tilde{E}(R, c_x, c_y)$ and $E(R, c_x, c_y)$ attain their minima are the same, and therefore, the proposed active contour model is invariant under global contrast image variation. On the other hand energy $E(R, c_x, c_y)$ is also independent of the circle size in the sense that term (11) is normalized with respect to circle boundary length and terms (12) and (13) are also normalized with respect to the integral domain sizes. Therefore, actually what really matters in the proposed energy to estimate the circle location is the quality of the circle rather than its size or its contrast with respect to the background.

4 Algorithm design

The proposed circle tracking algorithm can be divided into the following steps:

1. Initial circle location for the first frame.
2. For each image frame:
   (a) Compute the Gaussian convolution and the image gradient using GPU programming techniques.
   (b) Optimization of the circle location by minimizing energy (10).

The initial circle location for each frame is the one obtained in the previous frame. In the case of the first frame, the initial circle can be provided by hand or by any standard circle extraction method. In the experiments we present in this paper, we apply first the Hough transform

and then we select by hand the circle we are interested in. In the case of CT images, this selection can be automatic using other circle restrictions as the circle radius or that the circle is near to image center. In any case we point out that in this paper, we focus on the tracking step and the proposed method is independent of the initialization method used to extract the circle location in the first frame.

4.1 Evaluation of energy (10)

To evaluate numerically part (11) of energy (10), we just discretize the arc length taking a discretization step δ_L and we fix

$$N = round\left(\frac{2\pi R}{\delta_L}\right) \tag{19}$$

and then we approximate (11) using

$$\begin{aligned}\frac{1}{2\pi}\int_0^{2\pi} g(\nabla I_\sigma(C_{R,\bar{c}}(\theta)), C'_{R,\bar{c}}(\theta))d\theta \\ \approx \frac{1}{N}\sum_{n=0}^{N-1} g\left(\nabla I_\sigma\left(C_{R,\bar{c}}\left(\frac{n}{N}\right)\right), C'_{R,\bar{c}}\left(\frac{n}{N}\right)\right).\end{aligned} \tag{20}$$

We point out that in the case I be the characteristic function of the disk $D_R(c_x, c_y)$, if we fix $\sigma = 0$, then the above expression is just the contrast value between the disk and the background.

In order to evaluate numerically parts (12) and (13) of energy (10), we approximate the double integrals using a domain discretization like the one illustrated in Fig. 1. So we take δ_L and N as above and we denote by M the number of regions we use to discretize the domain in the radial direction, then we can approximate the double integral of any function $F(x, y)$ in the following way :

$$\int_0^{2\pi}\int_{R-\delta_R^-}^{R} F(C_{r,\bar{c}}(\theta))r\mathrm{d}r\mathrm{d}\theta \approx \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} F(C_{r_m,\bar{c}}(\theta_n))A_m,$$

where

$$\begin{cases} r_m = R - (m+0.5)\dfrac{\delta_R^-}{M} \\ \theta_n = \dfrac{n}{N} \\ A_m = \dfrac{\pi(R - \dfrac{m}{M}\delta_R^-)^2 - \pi(R - \dfrac{m+1}{M}\delta_R^-)^2}{N}, \end{cases} \tag{21}$$

we point out that we can use the same strategy to evaluate part (13) of the energy.

4.2 Energy optimization

We use a Newton–Raphson type algorithm to minimize energy (10). We denote by $\mathbf{u} = (R, c_x, c_y)$ the circle parameter vector we want to optimize. We use the following iterative scheme to minimize $E(\mathbf{u})$

$$\mathbf{u}_{n+1} = \mathbf{u}_n - (\nabla^2 E(\mathbf{u}_n) + \gamma Id)^{-1}\nabla E(\mathbf{u}_n), \tag{22}$$

where $\nabla^2 E(\mathbf{u}_n)$ is the 3×3 Hessian matrix, Id is the identity matrix, and $\nabla E(\mathbf{u}_n)$ is the gradient vector, γ is a damping parameter used to control the convergence of the minimization as follows: γ is updated in each iteration to ensure that $E(\mathbf{u}_{n+1}) < E(\mathbf{u}_n)$. Usually, its value is higher when we are far from the solution and decreases when we approach it. The idea of using a damping parameter in optimization algorithms was introduced by Levenberg in 1944 in the context of the well-known Levenberg–Marquardt optimization algorithm (see [28] for more details). In practice, the Hessian matrix and the gradient vector are computed using standard finite difference schemes.

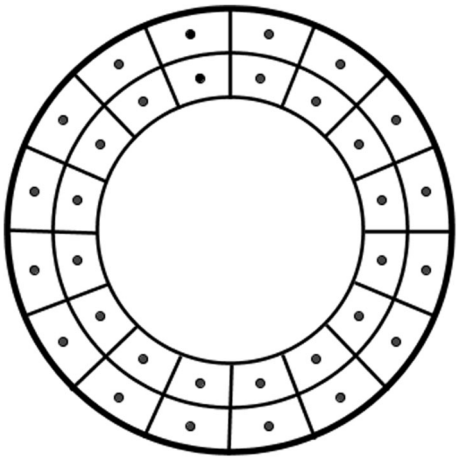

Fig. 1 Illustration of domain discretization to compute double integrals where $R = 2$, $\delta_L = 2\pi/16$, $\delta_R^- = \delta_L/2$, $M = 16$, and $N = 2$

5 Experimental results

In order to check the accuracy of the proposed method, we are going to first perform two experiments on synthetic image sequences. We have provided, as supplementary material of this paper, in the Online Resource 1 (FACTUACM_1.mpg) and the Online Resource 2 (FACTUACM_2.mpg), two videos to show these image sequences. In the first experiment, we use a sequence of 360 synthetic circles where the center location and radius change in a smooth way across the sequence with subpixel precision (the circle radius range in pixels is [5, 20]). In Fig. 2, we illustrate 2 frames of this image sequence. We compare the results of the proposed method with the ones obtained by the Hough transform. In each frame, Hough transform returns a collection of circles, to ensure that we always keep the best circle, we choose the circle with the

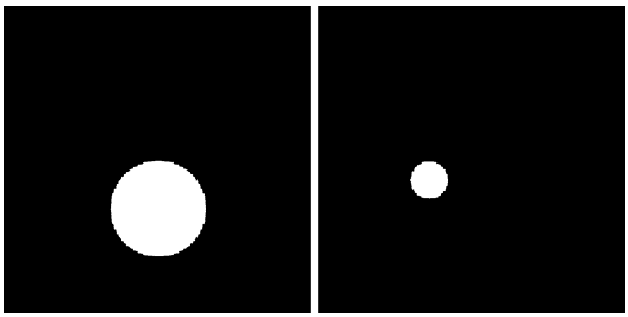

Fig. 2 Examples of two *circles* of the first synthetic image sequence we use for comparison purpose

Fig. 3 Examples of two *frames* of the second synthetic image sequence we use for comparison purpose

Table 1 Comparison results obtained by the Hough method and the proposed active contour model for the synthetic circle sequence illustrated in Fig. 2

	c_x	c_y	R
Average error Hough method	0.41	0.37	0.20
Average error proposed method	0.032	0.028	0.06

nearest center to the circle center of the previous frame and we allow a maximum variation of 2 pixels in the circle radius with respect to the previous circle. In this way we adapt, in a natural way, the Hough transform to track the evolution of a single circle and then, we can perform fair comparison between the Hough method and the proposed active contour model. In this paper, we use the Hough Transform implementation of the standard 4.3.2 ITK library using the following parameters: sweep angle 0–2π, with intervals of π/60 and maximum and minimum radius are equal to the radius obtained in the previous frame incremented/decremented by 2, respectively (in the first frame the radius is taken from the initial circle). ITK is a cross-platform, open-source application development framework widely used in computer vision applications.

In Table 1, we present some comparison results of the average error through the synthetic image sequence of the actual circle center location and radius and the ones obtained by both methods. We point out that the proposed method is much more accurate than the Hough transfom. We notice that to minimize energy (10), we have to fix 4 parameters: σ, the standard deviation of the Gaussian convolution kernel; α_-, α_+, the weights in the energy of the terms (12) and (13); and δ_R^-, which determines the sizes of the integral domains for terms (12) and (13). In most experiments we present in this paper we fix these parameters to $\sigma = 1$, $\alpha_- = \alpha_+ = 0.1$, and $\delta_R^- = 2$.

In the second synthetic experiment we present, we use the same circle sequence but we include in each image a white bar occluding partially the circle. In Fig. 3, we illustrate two frames of this sequence. In this experiment, when the size of the circle is reduced it becomes more difficult to extract the right circle from the image. The proposed active contour model finds the proper circle in all frames and in a more accurate way than the Hough transform as it is shown in Table 2.

The third experiment, illustrated in Fig. 4, corresponds to a real computed tomography (CT) image sequence composed of 416 frames. This technique is currently considered by the American College of Radiology (ACR) the most appropriate imaging modality for the correct diagnosis of several diseases of the aorta [1]. Previous to the acquisition, injection of an intravenous radiopaque contrast medium is performed in order to enhance contrast, following the standard procedures for obtaining such kind of images. The result is a pool of images stored in DICOM format files of 512 × 512 pixels with a resolution of 0.7 × 0.7 (mm/pixel) and 4096 gray levels. In this case, we want to track the evolution of the circle corresponding to the aorta vessel. In Fig. 6, we present some comparison results between the Hough method and the proposed active contour model method. We can observe in the figure that when there exist structures near the aorta vessel with high gradient magnitude the circle obtained by the Hough method can include points of such external structures which provide inaccurate results. We point out that in the context of medical aortic disease diagnosis, the accurate estimation of aorta diameter is a critical issue. In order to validate the results from a medical point of view, a radiologist has marked manually, for each frame, about 8 points in the aorta vessel contour. To improve the accuracy of the manual point selection, we use the image contour estimated by the subpixel edge detector introduced in [35]. Each time the radiologist marks a point by clicking in the image, we also compute the nearest contour point position in subpixel precision. In Fig. 5, we illustrate these manual marked points as well as their nearest contour points. The radiologist can accept or not the new position proposed by the

Table 2 Comparison results obtained by the Hough method and the proposed active contour model for the synthetic circle sequence illustrated in Fig. 3

	c_x	c_y	R
Average error Hough method	0.37	0.39	0.21
Average error proposed method	0.10	0.11	0.06

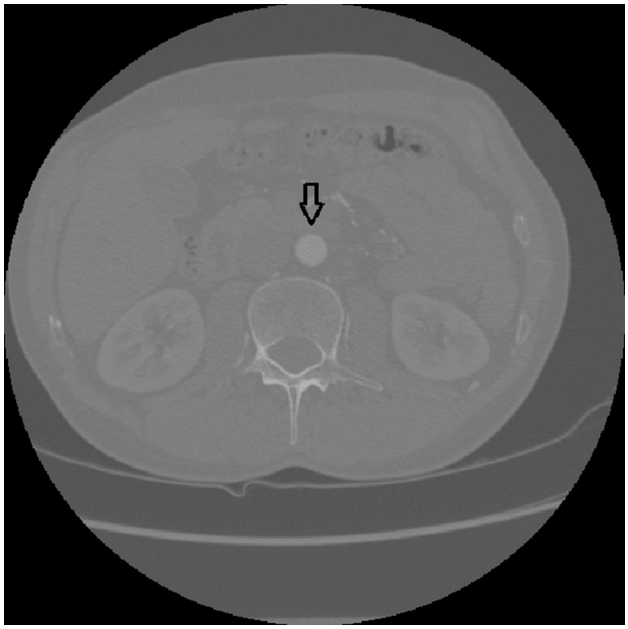

Fig. 4 Computed tomography (CT) image where we show the aorta location

contour detector. We point out that this manual procedure is completely independent of the method proposed in this paper and it is used just for medical validation purposes. We have included, in the Online Resource 3 (FACTUACM_3.mpg), a video where we show a zoom of the CT image centered in the aorta location where the aorta contour points selected by the radiologist are marked. To check the accuracy of the circle estimation methods for this sequence we compute the average through the image sequence of the error given by the distance of the manually marked points to the estimated circles. In Table 3, we present such error measure for the Hough method and the proposed active contour model. We point out that the proposed method is significantly more accurate than the Hough transform.

In Figs. 7, 8, and 9 we present some experiments on a real image sequence. In this sequence, a circle moves from a lighted to a darked area. Due to the poor light conditions, a number of problems appear including ISO noise in the dark areas, etc. In the Online Resource 4 (FACTUACM_4.mpg), we present a video with the image

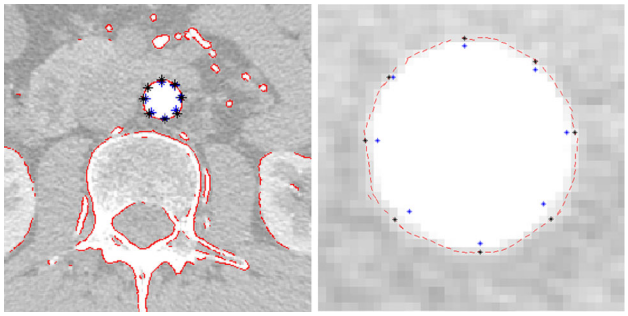

Fig. 5 Illustration of the manual identification of aorta contour points (original image on the *left* and detailed zoom on the *right*). *Blue marks* show the points marked by the radiologist (isolated points), and *black marks* show their closest edge points

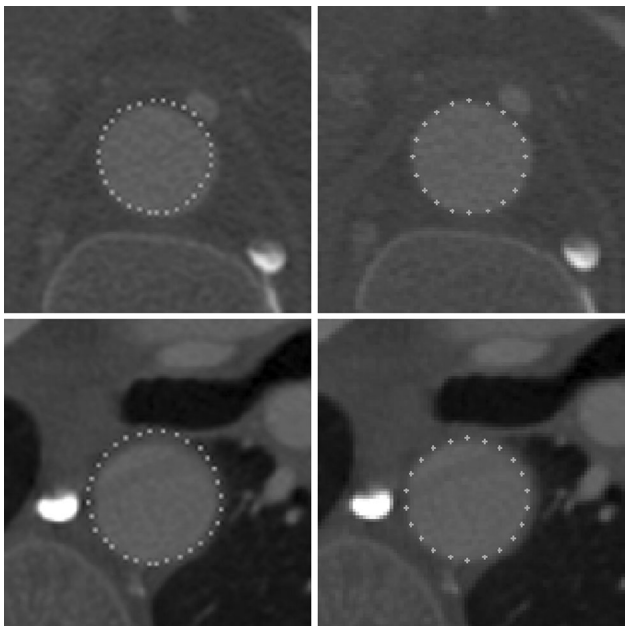

Fig. 6 Ilustration of the aorta circle tracking in different image frames. On the *left*, we present the circle (marked with *white dots*) obtained using the Hough method and on the *right*, the circle obtained using the proposed active contour model

Table 3 Comparison results obtained by the Hough method and the proposed active contour model for the CT image sequence

	Average distance error for CT image sequence
Hough method	0.766
proposed method	0.551

We compare the average distance (in pixels) between the manual marked points and the estimated circle for both methods

sequence where we show the results obtained with the proposed method. In Fig. 7, we present an experiment to show the ability of the proposed method to find out the correct circle even if we initialize the circle far away from

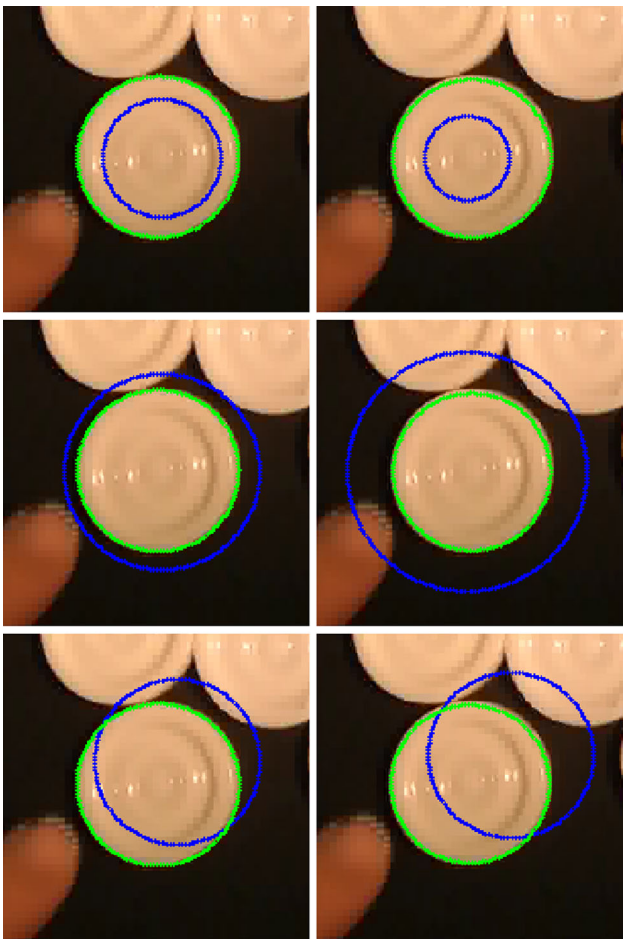

Fig. 7 We present some experiments using *different circle* initialization. In *blue* we present the *initial circle* and in *green* the final one obtained by minimizing energy (10). On the *left* we use $\sigma = 4$ as the Gaussian standard deviation parameter in energy (10), and on the *right* $\sigma = 8$

the correct one. The choice of the Gaussian standard deviation parameter σ in energy (10) determines how far we can initialize the circle, the larger σ, the farther we can initialize the circle because the Gaussian convolution filter enlarges the area of attraction of the correct circle when we minimize energy (10). In practice, the choice of σ depends on the expected circle velocity across the image sequence because in each frame we use as initialization of the circle the position in the previous frame. In Fig. 8, we present the results obtained in 2 frames of the sequence. We observe that the Hough transform is not able to detect the circles in the dark area because in such areas the edge detector fails to detect the circle contour. Since the proposed method does not use an edge estimation, it can recover properly the circle in the dark areas. In Fig. 9, we present the circle estimation result using the method proposed in [34]. Again, as the method uses an edge detector, it fails to recover the circle in the dark area.

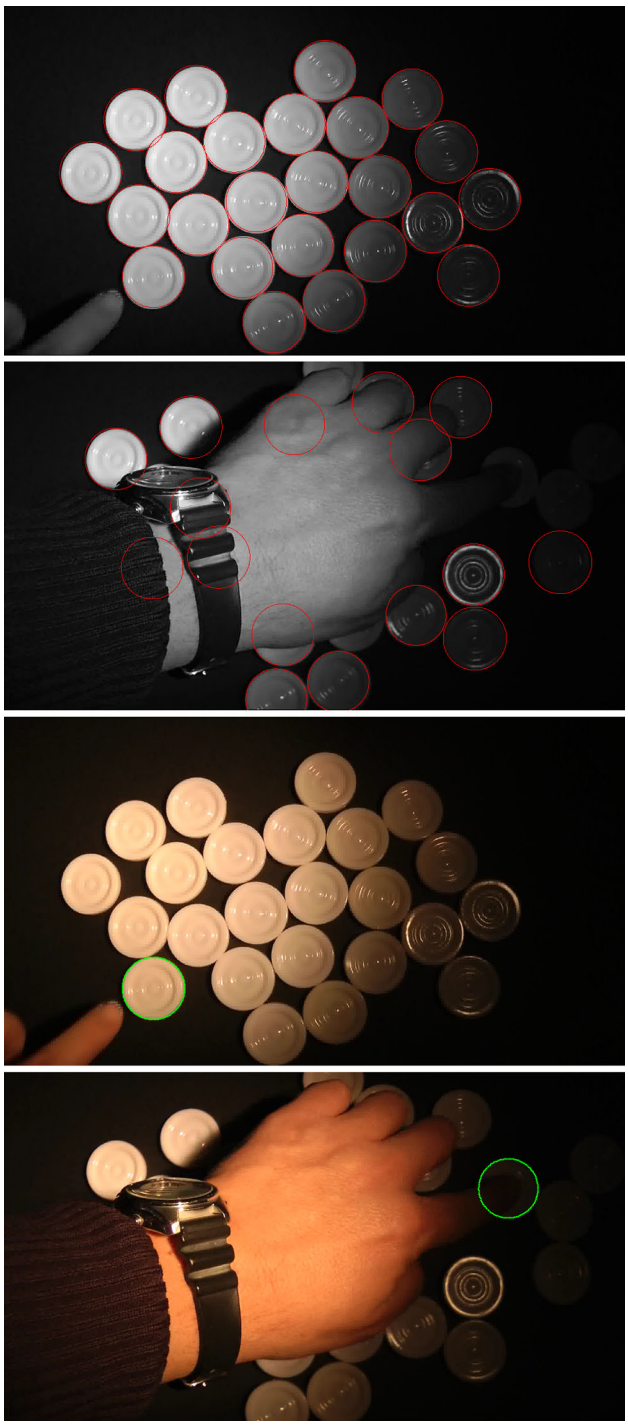

Fig. 8 Circle detection in two *frames* of a real image sequence. In the *top*, the results (in *red*) using the Hough transform. In the *bottom* (in *green*), the results using the proposed method

5.1 GPU-CUDA implementation

In order to optimize the circle location by minimizing energy (10), the proposed algorithm requires the computation of a Gaussian filter and two spatial derivatives (the

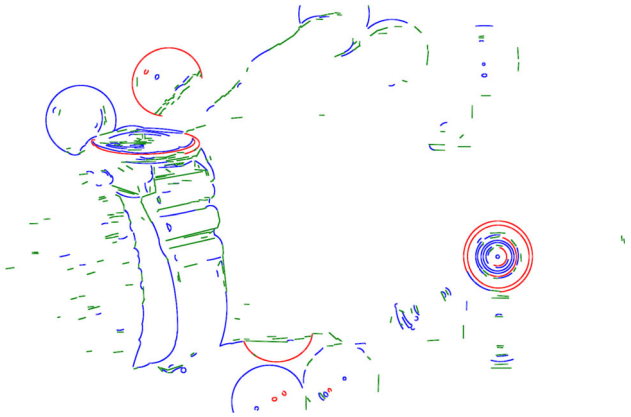

Fig. 9 Illustration of the *circles* obtained in an image of Fig. 8 using the method introduced in [34]

image gradient) for each frame of the sequence. The computation of such filters is the most time-consuming process of the algorithm. The CPU time for these calculations in a computer with an Intel I7 processor using a non-parallel implementation is about 65 % of the total runtime. This result gives us according to Amdahl's law an upper bound of the improvement factor equal to $1/0.35 \approx 2.9$ by only improving the runtime of the filters. To reduce the execution time, we use a GPU-CUDA implementation of the filters on a NVIDIA GTX 980 board with 2048 CUDA cores, 4 GB of GDDR5 memory, and CUDA 7.0.

The implementation of the algorithm is based on [33] adapted to use the radius of the kernel as an input variable and the new capabilities of the GPUs. This algorithm is mainly based on the use of the shared memory of the GPU by the threads within a block to reduce the memory latency of the global memory. The parameters to configure are the block size (`(BLOCKDIM_X, BLOCKDIM_Y)`) and the number of elements to compute for each thread `RESULT_STEPS`. These parameters can be configured independently to horizontal and vertical convolution. We also modified the algorithm in order to use the radius of the kernel as a template parameter, so the compiler generated in compile time the kernels for each radius with all unrolled loops. We checked the code using the branch efficiency metric provided by the NVIDIA tool `nvprof`, the result was zero in all cases meaning that the code had not any branches.

In order to tune the application, we followed the usual recommendations (see [11, 24, 30]): keep the number of threads per block a multiple of warp size (32), avoid small block sizes, adjust block size up or down according to kernel resource requirements, keep the number of blocks much greater than the number of SMs to expose sufficient parallelism to your device, and conduct experiments to discover the best execution configuration and resource usage.

We took into account the alignment of the 2D data in the global memory of the GPU (`cudaMallocPitch()`) and we also chose the horizontal dimension `x` of the block size to a multiple of the warp size in order to get coalesced access to memory. We used the command line tool `nvprof` to profile the execution of the CUDA code. Table 4 shows some results for a 512 × 512 image. The numbers in the configuration column are the x and y dimension of the block size and the number of results computed by thread. The column label "Achieved Occupancy" is the ratio of the average active warps per active cycle to the maximum number of warps supported on a multiprocessor. The global memory load efficiency (ratio of requested global memory load throughput to required global memory load throughput) was 100 % in all case. This result is explained by the choice of the horizontal dimension of the block size properly to get coalesced memory accesses.

According to Table 4, the best choice for the block size is (64, 8) where each thread compute 4 results (this configuration is a balance between a better use of the shared memory and a bigger number of blocks). When we run the algorithm with the best configuration, we obtained a execution total time of 13 ms, this is an improvement in a factor 2.2. The execution time can be divided in the execution time in the host (≈10 ms) and the execution time in the GPU (≈3 ms). The tool `nvprof` reports that the 70 % of the total GPU time is spent in transferring information between CPU and GPU.

We can also hide the GPU latency using the Asynchronous Streams features given by the CUDA API. For

Table 4 GPU implementation results of the Gaussian filter for a 512 × 512 image using a separable implementation and different configurations for the block size and number of results computed for each thread

Configuration	Rows Kernel (μs)	Achieved occupancy
(32, 32, 1)	19.263	0.905159
(32, 32, 2)	14.954	0.856547
(64, 8, 1)	16.705	0.899702
(64, 8, 4)	9.9470	0.897849
(32, 16, 1)	17.140	0.900978
(32, 16, 4)	12.040	0.866848
(16, 16, 1)	19.064	0.902142
(16, 16, 4)	16.754	0.845102

We show the results for the row direction convolution using a convolution mask of radius equals to 8. In the column direction the results are similar

Table 5 Computational time of the tracking procedure for a 12 × 512 image

	Computational time of the tracking procedure for a 12 × 512 image (ms)
CPU Hough method (ITK)	590
GPU Hough method [22]	73
Proposed method using a CPU non-parallel implementation	29
Proposed method using a GPU implementation	10

that, the algorithm launches a task in each Asynchronous Streams which include the transfer from CPU to GPU, the gaussian and the gradients and the transfer from GPU to CPU. All these tasks do not block the CPU and the tasks happen concurrently. This optimization allow us to hide the GPU latency and the total execution time was reduced to 10 ms, this mean a factor of 2.8 (near to the theoretical 2.9).

In Table 5, we show the computational time of the circle tracking estimation for a 512 × 512 image using the Hough transform (using the ITK implementation in a Intel(R) Core(TM) i7 2.80GHz computer), a GPU implementation of Hough transform [22], and the proposed active contour model. We point out that CPU computing time is shown just for illustration, we do not intend in this paper to compare CPU and GPU architectures.

6 Conclusions

In this paper, we propose an active contour model for circle tracking across an image sequence. Some interesting advantages of the method is that the associated energy is invariant under global contrast image variation and it does not depend on the circle size, moreover no edge estimation is required, energy computation is based just on the gradient of an image Gaussian kernel convolution which is computed using a GPU−CUDA implementation. The combination of an active contour model which does not requiere edge detection and a GPU–CUDA implementation provides a fast and accurate method for circle tracking. We present some experiments on synthetic image sequences which show that the proposed method is more accurate than the usual Hough transform. We validate the accuracy of the proposed method in the context of aorta vessel segmentation in computed tomography (CT). In this case, validation is performed by computing the average distance of the estimated circles with aorta contour points marked manually by a radiologist. We show that such average distance is much smaller using the proposed model than the Hough transform. We also present some experiments to show the ability of the proposed method to deal with poor light conditions. Since we do not use edge detector, the method is able to detect circles in very dark areas.

References

1. ACR Foundation: Acr appropriateness criteria (2014). https://acsearch.acr.org/list
2. Alemán-Flores, M., Alvarez, L., Caselles, V.: Texture-oriented anisotropic filtering and geodesic active contours in breast tumor ultrasound segmentation. J. Math. Imaging Vis. **28**(1), 81–97 (2007)
3. Alvarez, L., Baumela, L., Henriquez, P., Marquez-Neila, P.: Morphological snakes. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pp. 2197–2202 (2010)
4. Aubert, G., Kornprobst, P.: Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations, 2nd edn. Springer Publishing Company Incorporated, New York (2010)
5. Bascle, B., Deriche, R.: Features extraction using parametric snakes. In: Pattern Recognition, 1992. Vol. III. Conference C: Image, Speech and Signal Analysis, Proceedings., 11th IAPR International Conference on, pp. 659–662. IEEE (1992)
6. Brox, T., Kim, Y.J., Weickert, J., Feiden, W.: Fully-automated analysis of muscle fiber images with combined region and edge-based active contours. In: Handels, H., Ehrhardt, J., Horsch, A., Meinzer, H.P., Tolxdorff, T. (eds.) Bildverarbeitung fr die Medizin 2006, Informatik aktuell, pp. 86–90. Springer, Berlin (2006)
7. Brox, T., Rousson, M., Deriche, R., Weickert, J.: Colour, texture, and motion in level set based segmentation and tracking. Image Vis. Comput. **28**(3), 376–390 (2010)
8. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. Int. J. Comput. Vis. **22**(1), 61–79 (1997)
9. Catté, F., Lions, P.L., Morel, J.M., Coll, T.: Image selective smoothing and edge detection by nonlinear diffusion. SIAM J. Numer. Anal. **29**(1), 182–193 (1992)
10. Chan, T.F., Vese, L.A.: Active contours without edges. Trans. Imaging Proc. **10**(2), 266–277 (2001). doi:10.1109/83.902291
11. Cheng, J., Grossman, M., McKercher, T.: Professional CUDA C Programming. Wiley, Indianapolis (2014)
12. Davies, E.R.: The effect of noise on edge orientation computations. Pattern Recogn. Lett. **6**(5), 315–322 (1987)
13. De Fontes, F.P.X., Barroso, G.A., Coupé, P.: Real time ultrasound image denoising. J. Real-Time Image Process. **6**(1), 15–22 (2011)
14. Debreuve, E., Barlaud, M., Marmorat, J.P., Aubert, G.: Active Contour Segmentation with a Parametric Shape Prior: Link with the Shape Gradient. In: ICIP, IEEE, pp. 1653–1656 (2006)
15. Gadeski, E., Fard, H.O., Le Borgne, H.: Gpu deformable part model for object recognition. J. Real-Time Image Process., 1–13 (2014)
16. Havel, J., Dubská, M., Herout, A., Josth, R.: Real-time detection of lines using parallel coordinates and cuda. J. Real-Time Image Process. **9**(1), 205–216 (2014)
17. Herout, A., Josth, R., Juránek, R., Havel, J., Hradis, M., Zemcík, P.: Real-time object detection on cuda. J. Real-Time Image Process. **6**(3), 159–170 (2011)
18. Hough, P.: Method and means for recognizing complex patterns (1962). URL: http://www.google.co.in/patents/US3069654. Us patent 3,069,654

19. Illingworth, J., Kittler, J.: A survey of the hough transform. Comput. Vis. Graph. Image Process. **44**(1), 87–116 (1988)
20. Ioannou, D., Huda, W., Laine, A.F.: Circle recognition through a 2d hough transform and radius histogramming. Image Vis. Comput. **17**(1), 15–26 (1999)
21. Jacob, M., Blu, T., Unser, M.: Efficient energies and algorithms for parametric snakes. IEEE Trans. Image Process. **13**, 1231–1244 (2004)
22. Jan Essbach, B.L., Nacke, C.: Hough transform: Serial and parallel implementations. Tech. rep. URL: http://www.essbach.org/wp-content/uploads/2013/05/Hough
23. Kimme, C., Ballard, D., Sklansky, J.: Finding circles by an array of accumulators. Commun. ACM **18**(2), 120–122 (1975)
24. Kirk, D.B.: Programming Massively Parallel Processors: A Hands-on Approach, 1st edn. Morgan Kaufmann Publishers Inc., San Francisco (2010)
25. Kumar, P., Singhal, A., Mehta, S., Mittal, A.: Real-time moving object detection algorithm on high-resolution videos using gpus. J. Real-Time Image Process., 1–17 (2013)
26. Laborda, M.A.M., Moreno, E.F.T., del Rincón, J.M., Jaraba, J.E.H.: Real-time gpu color-based segmentation of football players. J. Real-Time Image Process. **7**(4), 267–279 (2012)
27. Lamas-Rodríguez, J., Heras, D.B., Arguello, F., Kainmueller, D., Zachow, S., Bóo, M.: Gpu-accelerated level-set segmentation. J. Real-Time Image Process., 1–15 (2013)
28. Levenverg, K.: A method for the solution of certain non-linear problems in least-squares. Q. Appl. Math. **2**(2), 164–168 (1944)
29. Marquez-Neila, P., Baumela, L., Alvarez, L.: A morphological approach to curvature-based evolution of curves and surfaces. IEEE Trans. Pattern Anal. Mach. Intell. **36**(1), 2–17 (2014)
30. NVIDIA, C.: Cuda c best practices guide. Technical report. URL: http://docs.nvidia.com/cuda/cuda-c-best-practices-guide
31. Pao, D.C.W., Li, H.F., Jayakumar, R.: Shapes recognition using the straight line hough transform: theory and generalization. IEEE Trans. Pattern Anal. Mach. Intell. **14**(11), 1076–1089 (1992)
32. Paragios, N., Deriche, R.: Geodesic active regions and level set methods for motion estimation and tracking. Comput. Vis. Image Underst. **97**(3), 259–282 (2005)
33. Podlozhnyuk, V.: Image convolution with cuda. NVIDIA Corporation, Technical report (2007)
34. Ptrucean, V., Gurdjos, P., von Gioi, R.: A parameterless line segment and elliptical arc detector with enhanced ellipse fitting. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) Computer Vision ECCV 2012. Lecture Notes in Computer Science. Springer, Berlin Heidelberg, pp. 572–585 (2012)
35. Trujillo-Pino, A., Krissian, K., Aleman-Flores, M., Santana-Cedres, D.: Accurate subpixel edge location based on partial area effect. Image Vision Comput. **31**(1), 72–90 (2013)
36. Tsuji, S., Matsumoto, F.: Detection of ellipses by a modified hough transformation. IEEE Trans. Comput. **27**(8), 777–781 (1978)
37. Ujaldon, M., Ruiz, A., Guil, N.: On the computation of the circle hough transform by a GPU rasterizer. Pattern Recogn. Lett. **29**(3), 309–318 (2008)
38. Wang, Y.K., Huang, W.B.: A cuda-enabled parallel algorithm for accelerating retinex. J. Real-Time Image Process. **9**(3), 407–425 (2014)

Carmelo Cuenca received his M.Sc. in Physics in 1987 from Complutense University of Madrid (UCM) and in 2004 received his Ph.D. in Computer Science from the University of Las Palmas de Gran Canaria (ULPGC). Currently, he is an Associate Professor at ULPGC. His main research areas are Computer Vision and algorithm parallelization using GPU.

Esther González received her M.Sc. in Computer Science in 1991 and Ph.D. in Computer Science in 2008, both from University of Las Palmas de Gran Canaria (ULPGC). Currently, she is an Associate Professor at ULPGC. Her main research areas are Computer Vision and Biometry.

Agustín Trujillo received his M.Sc. in Computer Science in 1991 and Ph.D. in Computer Science in 2004, both from University of Las Palmas de Gran Canaria (ULPGC). Currently, he is an Associate Professor at ULPGC. His main research areas are Computer Vision and Computer Graphics.

Julio Esclarín received his M.Sc. in Mathematics in 1979 from Zaragoza University (Spain), and in 1996 received his Ph.D. in Computer Science from the University of Las Palmas de Gran Canaria (ULPGC). Currently, he is an Associate Professor at ULPGC. His main research areas are Mathematical Analysis and Computer Vision.

Luis Mazorra received his M.Sc. in Mathematics in 1980 from La Laguna University (Spain) and in 1994 received his Ph.D. in Computer Science from the University of Las Palmas de Gran Canaria (ULPGC). Currently, he is an Associate Professor at ULPGC. His main research areas are Mathematical Analysis and Computer Vision.

Luis Alvarez received his M.Sc. in Mathematics in 1985 and Ph.D. in Applied Mathematics in 1988, both from Complutense University (Madrid, Spain). Between 1991 and 1992 he worked as post-doctoral researcher at CEREMADE laboratory, Université Paris IX (Dauphine) (France). Since 2000, he is full Professor in Computer Sciences at Universidad de Las Palmas de Gran Canaria. His main research areas are Partial Differential Equations and Computer Vision.

Juan Antonio Martínez-Mera received a M.Sc. in Physics in 2008 and a Ph.D. in Physics in 2014, bothfrom University of Santiago de Compostela (USC). Currently, he has a postdoc position at USC. His mainresearch areas are Computer Vision and Medical Imaging.

Pablo G. Tahoces received his M.Sc. in Physics in 1988 and Ph.D. in Physics in 1992, both from University of Santiago de Compostela (USC). Currently, he is an Associate Professor at USC. His main research areas are Computer Vision and Medical Imaging.

José M. Carreira received his M.Sc. in Medicine and Surgery in 1983 and Ph.D. in Medicine and Surgery in 1991, both from University of Santiago de Compostela (USC). Currently, he is an associate professor at USC. He also works as radiologist at University Hospital Complex of Santiago de Compostela. His main research areas are Computer Aid Diagnosis Systems and Vascular and Interventional Radiology.