Computer Vision and Image Understanding 151 (2016) 29–46

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

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

Adaptive particle filtering for coronary artery segmentation from 3D CT angiograms

David Lesage [a,1], Elsa D. Angelini [b,c], Gareth Funka-Lea [d], Isabelle Bloch [b,*]

[a] *Philips Research, Suresnes, France*
[b] *LTCI, CNRS, Télécom ParisTech, Université Paris Saclay, Paris, France*
[c] *Columbia University, Department of BME, New York, NY, USA*
[d] *Medical Imaging Technologies, Siemens Healthcare, Princeton, USA*

ARTICLE INFO

Article history:
Received 13 February 2015
Accepted 18 November 2015

Keywords:
3D CTA
Coronary segmentation
Bifurcations
Tracking with particle filter
Bayesian model
Geometric model
Flux feature
Kernel estimation
Sampling scheme
Mean-shift

ABSTRACT

Considering vessel segmentation as an iterative tracking process, we propose a new Bayesian tracking algorithm based on particle filters for the delineation of coronary arteries from 3D computed tomography angiograms. It relies on a medial-based geometric model, learned by kernel density estimation, and on a simple, fast and discriminative flux-based image feature. Combining a new sampling scheme and a mean-shift clustering for bifurcation detection and result extraction leads to an efficient and robust method. Results on a database of 61 volumes demonstrate the effectiveness of the proposed approach, with an overall Dice coefficient of 86.2% (and 92.5% on clinically relevant vessels), and a good accuracy of centerline position and radius estimation (errors below the image resolution).

1. Introduction

In biomedical applications, vascular structures are often of critical importance for diagnosis, treatment and surgery planning. Vessels are thin, elongated and complex structures embedded in increasingly large images. Manual delineation, although still heavily used in clinical routines, has become a considerable burden and automatic or semi-automatic segmentation remains challenging.

Vascular segmentation has received considerable attention in the literature [41]. A popular approach is to consider the segmentation as an iterative, *tracking* process. Classical region-growing techniques can be seen as primitive representatives of this class of methods. Front propagation techniques allow for a refined analysis by imposing a structurally coherent exploration process. The robustness of local deterministic tracking is generally limited by the necessity of using low-level causal criteria. In some settings, the tracking problem has been formulated as the extraction of globally optimal paths [11,43,46]. Another approach, which is increasingly popular, is the use of stochastic Bayesian tracking algorithms such as particle filters [1,16,17,42,49,52,54,55,62–64,68]. Such algorithms have demonstrated particular robustness while allowing for high-level modeling.

In this paper, we propose a new Bayesian, stochastic tracking algorithm for the delineation of coronary arteries from 3D Computed Tomography Angiograms (CTA). Our approach is inspired by recent developments in particle filtering designs [1,16,17,42,52,54]. It relies on a medial-based geometric model and on a simple, fast and discriminative flux-based image feature [44], described in Section 3. The proposed method includes the following contributions:

- the design of a geometric vascular model described in Section 2;
- the introduction of a non-parametric Bayesian model, learned by kernel density estimation [56] from a ground-truth database of manually segmented datasets (Section 4);
- the design of a new sampling scheme, *Adaptive Auxiliary Particle Filtering* (AAPF), described in Sections 6 and 7 after briefly recalling the bases of particle filters in Section 5;
- the use of mean-shift clustering [9,21] for bifurcation detection and coronary tree extraction, along with the proposal of algorithmic refinements for increased computational efficiency (Section 7).

[*] Corresponding author. fax: +33 1 45 81 37 94.
E-mail address: isabelle.bloch@telecom-paristech.fr, isabelle.bloch@enst.fr (I. Bloch).

[1] This work was performed during David Lesage's Ph.D. thesis at Telecom ParisTech and Siemens Corporation, Corporate Research, Imaging & Computer Vision.

http://dx.doi.org/10.1016/j.cviu.2015.11.009
1077-3142/© 2015 Elsevier Inc. All rights reserved.

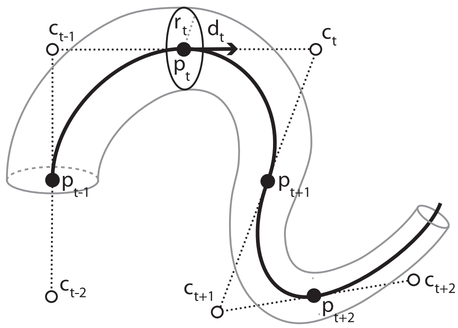

Fig. 1. Discrete medial-based geometric model (see text for notations).

A series of experiments is presented in Section 8, illustrating theoretical and practical properties of our approach, along with qualitative and quantitative evaluation on clinical data.

2. Geometric model

In this work, we chose to model vascular structures using a medial representation inspired by general shape models such as the Medial Axis Transform (MAT) from [4,5] and the Smoothed Local Symmetry (SLS) model from [6]. The main idea behind medial models applied to 3D elongated structures such as vessels is to represent the shapes of interest through their main axis, the *centerline* curve lying at the center of the vessel (Fig. 1).

We combine centerline- and cross-section-based information to constrain and reduce the parameter space with a particular discrete parameterization, illustrated in Fig. 1. Cross-sections, defined in locally orthogonal planes along the curved centerline, are assumed to be circular. This hypothesis is reasonable for the description of small scale vessels such as coronary arteries and enables straightforward parameterization. The centerline curve is discretized as a series of centerline points $\{p_t\}_{t=0,\dots,L}$, with associated radius values and tangential direction vectors, noted $\{r_t\}$ and $\{d_t\}$, respectively. Radius values and tangent directions define cross-sectional contours. A vascular segment is modeled as a series of triplets $x_{0:L} = \{(p_t, r_t, d_t)\}_{t=0,\dots,L}$. Individual elements $x_t = (p_t, r_t, d_t)$ are used as the *state* variables of the vessel model being optimized during the tracking process. We assume an order on the states, denoted by subscripts $t \in [\![0, L]\!]$. For coronary arteries, a natural ordering is from the ostium x_0 (origin of the artery branching off the aorta) to their distal ends x_L. Tangential directions $\{d_t\}$ are defined thanks to *control* points $\{c_t\}$: $d_t = \frac{c_t - c_{t-1}}{\|c_t - c_{t-1}\|}$.

To further constrain our geometric model, we propose to link the positions of centerline and control points, indirectly coupling centerline points and tangent directions, as:

$$p_t = \frac{c_t + c_{t-1}}{2} \tag{1}$$

This scheme, closely related to cardinal spline models, can be viewed as an artificial parameterization simplifying the formulation of our model and reducing its dimensionality. It makes possible a stable definition of tangential directions even in areas of high curvature. By doing so, control points constrain both the definition of tangential directions and the discretization of the centerline curve. States $x_t = (p_t, r_t, d_t)$ of our model can be described alternatively as $x_t = (c_{t-1}, c_t, r_t)$ given control points and radiuses, both being equivalent. The overall dimensionality of our model is thus limited to 4D (3D control point locations + radius values).

By convention, we consider that the first centerline point p_0 is fixed and that the first tangential direction d_0 is defined solely by c_0. It is equivalent to considering an implicit control point $c_{-1} = 2p_0 - c_0$. Whenever needed, centerline points and corresponding tangents can be used to conveniently interpolate the centerline curve, *e.g.* using cubic Hermite splines.

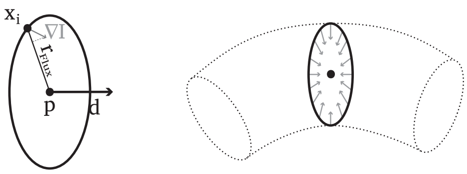

Fig. 2. `Flux` image feature. Discretized cross-sectional pattern defined by parameters (p, r_{flux}, d), with $r_{\mathtt{Flux}}$ the *test* radius. For each x_i, the gradient vector $\nabla I(x_i)$ is projected on the inward radial direction, $u(x_i) = \frac{p - x_i}{\|p - x_i\|}$.

One key parameter of our model is the spacing between successive control points $s = \|c_{t-1} - c_t\|$. This discretization step directly impacts the expressive power of the model. As it gets smaller, the model is able to depict accurately highly curved vessels. In this work, we used a fixed discretization step of the order of the data intra-slice resolution (0.3mm) to provide an accurate description of typical coronary arteries.

3. Flux-based vessel-dedicated feature

To feed our geometric model with image information, we employ a fast, discriminative image feature, referred to as `MFlux` [44]. This feature exploits *gradient flux* for the detection of elongated structures with circular cross-sections.

As demonstrated in [12,31,35,60], flux-based segmentation methods are well adapted for the extraction of thin, low-contrast vessels. They exploit the orientation of the gradient vectors by computing the gradient flux through the surface of the extracted object. For CTA images, we assume that vessels are hyper-intense, and maximize the *inward* flux through the circular *cross-sections* of the model. For slowly narrowing or widening vessels, the radial directions give a reasonable approximation of the local normals to the surface (see Fig. 2). After equi-angular discretization of the cross-section (orthogonal to d) perimeter with radius r into N points x_i, we obtain the following cross-sectional flux measure:

$$\mathtt{Flux}(p, r, d) = \frac{1}{N} \sum_{i=1}^{N} \langle \nabla I(x_i), u_i \rangle \tag{2}$$

with $\nabla I(x_i)$ the gradient vector at point x_i and $u_i = \frac{p - x_i}{|p - x_i|}$ the inward radial direction as defined in Fig. 2. Being a linear feature, *Flux*(p, r, d) is prone to false positive high-values at step-edges, as already mentioned by [38]. In our case, this behavior is particularly problematic along the heart chambers. A non-linear combination was therefore proposed to pair diametrically opposed points (x_i, x_i^{π}) and retain the *minimal* flux contribution per pair, similarly to what was done in 2D by [38]. The `MFlux` feature is defined as:

$$\mathtt{MFlux}(p, r, d) = \frac{2}{N} \sum_{i=1}^{\frac{N}{2}} \min(\langle \nabla I(x_i), u_i \rangle, \langle \nabla I(x_i^{\pi}), u_i^{\pi} \rangle)$$

with $x_i^{\pi} = x_{\frac{N}{2}+i}$ for an even number N of cross-sectional points.

The implementation of `MFlux` is particularly straightforward and computationally efficient. In the present work, we used $N = 8$ cross-sectional points and employed tri-linear interpolation for the computation of image gradient vectors.

`MFlux` responses are used as *image features* and combined with model-based prior knowledge, within the Bayesian tracking model described in the next section.

4. Bayesian vessel model

Our geometric model defines a vessel as a discrete ordered chain of states $x_{0:L} = \{(p_t, r_t, d_t)\}_{t=0,\dots,L}$ with p_t the centerline points, r_t the radius values and d_t the local tangential directions. A particular chain

$x^i_{0:L} = \{(p^i_t, r^i_t, d^i_t)\}_{t=0,\dots,L}$ can be seen as a *realization* of the model, indexed by i, considered as a random variable. A classical way of formulating a segmentation or detection task is through the maximization of the posterior probability of the model. The Maximum A Posteriori (MAP) problem corresponds to the extraction of the most probable realization given the data:

$$x^*_{0:L} = \arg\max_{\mathcal{X}_L} p(x_{0:L}|z_{1:L}) \tag{3}$$

where $z_{1:L} = \{z_t\}_{t=1,\dots,L}$ are observations available at each step t and $\mathcal{X}_L$ is the realization space. Our application differs from a classical MAP problem. In our case, the length L of the target vessel is unknown and there might be several vessels of interest (branching structures). Also, given the variability of vascular networks, our model covers a large parameter search space which explodes exponentially with the length of the chain. In a broad sense, our approach uses estimations of the posterior probability distribution to sample, compare and select model realizations.

The posterior distribution is given by Bayes' rule:

$$p(x_{0:L}|z_{1:L}) = \frac{p(x_{0:L})p(z_{1:L}|x_{0:L})}{p(z_{1:L})} \tag{4}$$

where $p(x_{0:\,L})$ is the joint *prior* distribution and $p(z_{1:\,L}|x_{0:\,L})$ the joint *likelihood* of the observations. The prior probability is a model-based term, which integrates prior knowledge on the variations of the model. The likelihood term can be thought of as a data fidelity term.

Assuming a first order Markovian transition model[2], the joint prior can be expressed as:

$$p(x_{0:L}) = p(x_0)\prod_{t=1}^{L} p(x_t|x_{t-1}) \tag{5}$$

If we further assume that the observation sets $\{z_t\}$ are conditionally independent given the model, and that each observation set z_t only depends on the state x_t at t, we have:

$$p(z_{1:L}|x_{0:L}) = \prod_{t=1}^{L} p(z_t|x_t) \tag{6}$$

These assumptions are classically adopted for sequential Bayesian estimation problems [15]. In particular, they allow for the formulation of the posterior probability in a recursive form.

Combining Eqs. (4)–(6), one can derive a recursive form of the posterior probability [15] until step t:

$$\begin{aligned} p(x_{0:t}|z_{1:t}) &= p(x_0)\frac{\prod_{k=1}^{t} p(x_k|x_{k-1})p(z_k|x_k)}{p(z_{1:t})} \\ &= \frac{p(x_t|x_{t-1})p(z_t|x_t)}{p(z_t|z_{1:t-1})}p(x_{0:t-1}|z_{1:t-1}) \end{aligned} \tag{7}$$

The posterior probability is updated from $t-1$ to t through the transition prior $p(x_t|x_{t-1})$ and marginal likelihood $p(z_t|x_t)$.

In the next sections, we detail how each term is estimated.

4.1. Learning from a ground-truth database: general scheme

We exploit a ground-truth database of 61 manually delineated cardiac CTA datasets, totaling 858 arteries with centerline and radius segmentation from experts. An example of such a ground truth delineation is given in Fig. 3. Our database provides us with samples of various statistical variables used by our Bayesian model, including successive scales, directions and associated `MFlux` responses. We exploit this valuable source of statistical information by learning the distributions composing Eq. (7). Most related works [1,16,17,42,52,54,63] rely on parametric distributions for this learning step. We prefer non-parametric density estimation techniques, also used in [62], to estimate vessel likelihood distributions from multiple image cues. In our work, non-parametric kernel estimation is used extensively, from learning likelihood terms (both vessel and background distributions) to direction and radius transition priors.

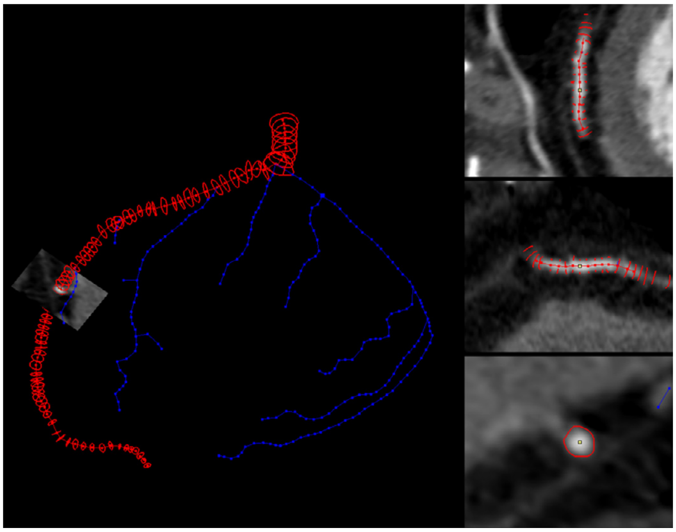

Fig. 3. Manual delineation of coronary vessel trees, obtained by specifying centerline locations in the vessel lumen. Corresponding cross-sectional contours were obtained automatically and corrected manually when needed. Left: 3D view of a full coronary tree, with one highlighted branch (in red). Right, from top to bottom: two cross-sectional and one axial MPR view along the delineated branch. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

Considering two random variables Z_1 and Z_2, our general estimation scheme can be described as follows:

1. estimate the joint density $\hat{f}(Z_1, Z_2)$ from samples $\{(Z^i_1, Z^i_2)\}$;
2. extract conditional probability densities of the form $p(Z_2|Z_1 = z_1)$.

An overview of this density estimation scheme is given in Fig. 4. The density $\hat{f}(Z_1, Z_2)$ captures the joint variations of Z_1 and Z_2 (*e.g.* joint variations of successive radiuses with $Z_1 = r_{t-1}$ and $Z_2 = r_t$). We learn these joint densities from the ground-truth database, using non-parametric kernel estimation [56]. Given these joint densities, we can eventually extract any conditional probability distribution needed by our model. This general principle was applied to both the likelihood and prior parts of our model.

For the sake of simplicity, we empirically chose a two-dimensional, uncorrelated Gaussian kernel [56]. The estimated joint density $\hat{f}(Z_1, Z_2)$ of two random variables Z_1 and Z_2 is given by:

$$\hat{f}(Z_1, Z_2) = \frac{1}{nh_1h_2}\sum_{i=1}^{n} K_g\left(\frac{Z_1 - Z^i_1}{h_1}, \frac{Z_2 - Z^i_2}{h_2}\right)$$

with $K_g(x, y) = \frac{1}{2\pi}\exp(-\frac{1}{2}(x^2 + y^2))$ where (Z^i_1, Z^i_2) are n samples from our database, and h_1 and h_2 are bandwidth parameters associated with Z_1 and Z_2, respectively.

Continuous joint densities are therefore learned from our database of discrete samples. This estimation is said to be non-parametric as no distribution models are imposed. In the following, the bandwidth parameters h_1 and h_2 are specified empirically, depending on the variables Z_1 and Z_2 considered. They are always chosen so as not to over-smooth the resulting densities and preserve the subtle variations embedded in the training database.

Considering the joint density $\hat{f}(Z_1, Z_2)$ as a joint probability density $p(Z_1, Z_2)$ (after normalization), we can extract any conditional

[2] Note that for our application, a first order model is sufficient. A 2nd order model would enforce curvature constraint, which is delicate since changes in curvature appear spuriously and suddenly. Most importantly, a higher order model would significantly increase the overall complexity of the particle filtering scheme.

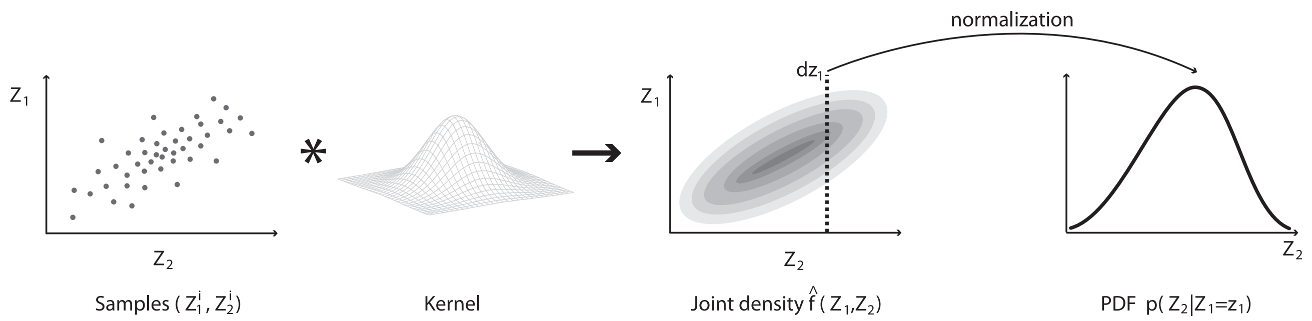
normalization
dz$_1$
Z$_1$
Z$_2$
Z$_1$
Z$_2$
Z$_2$
Samples (Z$_1^i$, Z$_2^i$)
Kernel
Joint density $\hat{f}$(Z$_1$,Z$_2$)
PDF p(Z$_2$|Z$_1$=z$_1$)

Fig. 4. Overview of the density estimation process.

probability density function (PDF) of the form $p(Z_2|Z_1=z_1)$ following Bayes' rule:

$$\begin{aligned} p(Z_2|Z_1=z_1) &= \frac{p(Z_2, Z_1 \in dz_1)}{p(Z_1 \in dz_1)} \\ &= \frac{p(Z_2, Z_1 \in dz_1)}{\int p(Z_2 \in dz_2, Z_1 \in dz_1) dz_2} \end{aligned} \tag{8}$$

The notation $p(Z_2|Z_1=z_1)$ indicates that conditioning is performed with respect to a given realization z_1 of Z_1. Notations dz_1 and dz_2 correspond to infinitesimal volumes around realizations z_1 and z_2 on which continuous distributions are locally integrated.

We now apply this approach to the estimation of $p(z_t|x_t)$ and $p(x_t|x_{t-1})$.

4.2. *Marginal likelihood* $p(z_t|x_t)$

The term $p(z_t|x_t)$ corresponds to the joint likelihood of the available observations, given the model state at time t. At each step t, $z_t = \{y_t^j\}_{j=1,\ldots,N_t}$ is the set of available observations. The number of observations, N_t, corresponds to the number of particles, *i.e.* the number of samples used to estimate the target distribution in our method (see Section 5).

We assume that observations y_t^j are conditionally independent given the model:

$$p(z_t|x_t) = \prod_{j=1}^{N_t} p(y_t^j|x_t) \tag{9}$$

Observations $\{y_t^j\}$ are typically obtained as image measurements for model state realizations $\{x_t^i\}_{i=1,\ldots,N_t}$ (see Section 3). In effect, one manipulates coupled sets $\{(x_t^i, y_t^i)\}_{i=1,\ldots,N_t}$ where observations are associated with model realizations. In the following, we use different index notations (j versus i) to emphasize the general case where a given observation y_t^j is not associated with the particular state realization x_t^i being considered ($j \neq i$). We simply note y_t^i the specific observation associated with the state realization x_t^i (same index, $j = i$).

Following the principles proposed in [22,39,66,67], a particularity of our observation model is to consider both *vessel* and *background* likelihood distributions. Most works in the literature do not express any *a priori* on the background distribution, which is equivalent to assuming, implicitly, a uniform background. In contrast, we exploit prior knowledge on non-vessel likelihood distributions. Observations can either be generated by actual vessels with distribution $p_v(y_t|x_t)$, or by the background with distribution $p_{bg}(y_t)$.

Given a specific state realization x_t^i, the likelihood of an observation y_t^j is evaluated as follows:

$$p(y_t^j|x_t = x_t^i) = \begin{cases} p_v(y_t^i|x_t = x_t^i) & \text{if } i = j \\ p_{bg}(y_t^j) & \text{otherwise} \end{cases} \tag{10}$$

If $j = i$, that is, if the observation $y_t^j = y_t^i$ is associated with the state realization x_t^i, it is considered as coming from a vessel distribution with probability $p_v(y_t^i|x_t = x_t^i)$. If $j \neq i$, *i.e.*, the observation y_t^j is not associated with the considered realization x_t^i, the likelihood is valued as coming from the background distribution with probability $p_{bg}(y_t^j)$. We emphasize that vessel likelihood PDFs $p_v(y_t|x_t)$ are dependent on the current model geometry (its radius r_t). The background distribution $p_{bg}(y_t)$ is considered independent of the model geometry.

For a specific realization x_t^i, one can now decompose the marginal likelihood for the set of observations z_t:

$$p(z_t|x_t = x_t^i) = p_v(y_t^i|x_t = x_t^i) \prod_{j \neq i} p_{bg}(y_t^j) \tag{11}$$

From Eq. (11), one can see that the valuation of the likelihood is dependent on the model state realizations. In other words, different model realizations lead to different interpretations of the set of observations. In that sense, realizations compete for the best explanation of all the observations. One can re-factor Eq. (11) as follows:

$$p(z_t|x_t = x_t^i) = \frac{p_v(y_t^i|x_t = x_t^i)}{p_{bg}(y_t^i)} \prod_{y_t^j \in z_t, j=1,\ldots,N_t} p_{bg}(y_t^j) \tag{12}$$

The first term in Eq. (12) is a likelihood ratio between vessel and background for the single observation y_t^i linked to the state realization x_t^i. This ratio encodes the competition between vessel and background distributions for the interpretation of the observation. The second term is a product over all the observations composing $z_t = \{y_t^i\}_{i=1,\ldots,N_t}$ and is independent of the model state. This allows us to focus solely on the observations associated with the model realization at hand.

4.2.1. *Vessel likelihood* $p_v(y_t|x_t)$

Observations y_t are assumed to be invariant with respect to rotations and translations, so that $p_v(y_t, x_t) = p_v(y_t, r_t)$. We estimate the joint density $\hat{f}_v(y_t, r_t)$, with the joint density of `MFlux` responses for true vessels over different radiuses (Fig. 5).

From the joint density, we can extract any likelihood PDF of the form $p_v(y_t|x_t = x_t^i) = p_v(y_t|r_t = r_t^i)$ by considering a particular radius realization r_t^i. A selection of such density functions is illustrated in Fig. 7. Our non-parametric density estimation is able to capture subtle variations of `MFlux` responses as a function of the target radius. In particular, one can note in Fig. 7 that responses tend to be lower, and of larger variance, for smaller vessels, which can be explained by the loss of contrast affecting small-scale coronary arteries.

4.2.2. *Background likelihood* $p_{bg}(y_t)$

To derive the background likelihood $p_{bg}(y_t)$, we exploit ideas first introduced in [39]. For a given cardiac CTA dataset, we assume that randomly selected parameters are unlikely to correspond to coronary arteries. Random sampling is thus employed to learn *dataset-specific* background distributions.

Given a dataset, we randomly sample 10^5 parameters (p_t^i, r_t^i, d_t^i) for which we evaluate `MFlux` responses. This number proved to

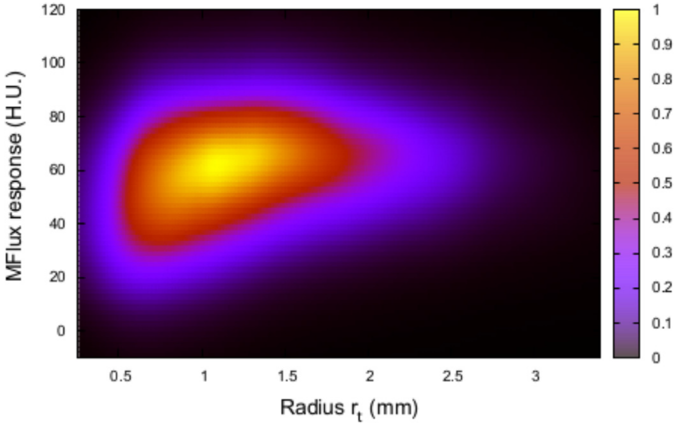

Fig. 5. Joint density $\hat{f}_v(y_t, r_t)$ of observations y_t (`MFlux` responses) for true vessels over radiuses r_t. Density estimation was performed on samples (y_t^i, r_t^i) from 10 randomly selected datasets, using kernel estimation with bandwidth parameters $h_{y_t} =$ 10 H.U. (Hounsfield Units) and $h_{r_t} = 0.3$ mm.

be large enough to provide us with stable statistics. Our sampling scheme excludes hypo-intense areas such as lungs. Indeed, given the intrinsic calibration of CT Hounsfield values, very hypo-intense values (typically, air) can be directly discarded without ambiguity. Our background model focuses on the ambiguous intensity range (soft tissues and lightly injected myocardium) typically observed in the surroundings of target arteries, with the aim of separating actual arteries from their immediate surroundings. Discarding hypo-intense from background sampling can be seen as a simple way to bias the model in order to focus it on the truly informative range. We use a Gaussian kernel of standard deviation $\sigma_{y_t} =$ 10H.U. to estimate the continuous likelihood PDF $p_{bg}(y_t)$ from the histogram of `MFlux` responses.

We assume that the distribution $p_{bg}(y_t)$ is independent of the model realization. As illustrated in Fig. 6, this assumption holds well over the range of considered radiuses. Learning dataset-specific background likelihood distributions allows us to capture subtle variations between datasets. As can be seen in Fig. 7, background likelihood distributions vary mainly with the image noise level.

Background responses correspond, as expected, to low `MFlux` values, whereas vessel likelihood distributions concentrate on high values (Fig. 7). The good separability of background and vessel likelihood distributions confirms the high discriminative power of the underlying `MFlux` feature. Overlaps between those distributions implicitly encode remaining possible ambiguities, *e.g.* between low-contrast vessels and small, bright non-vascular structures.

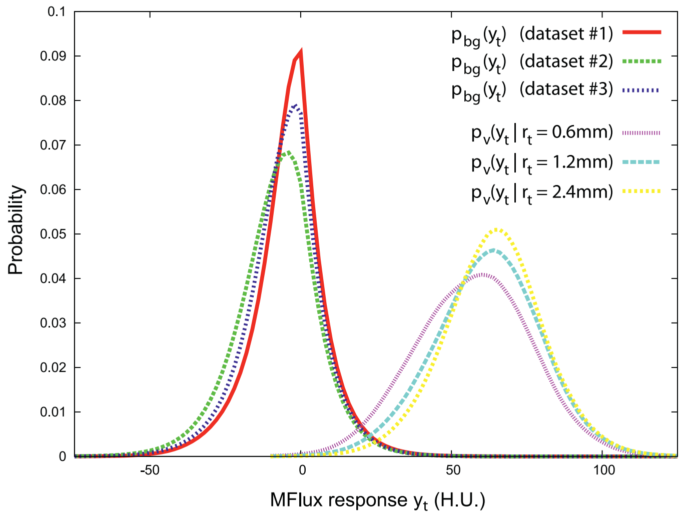

Fig. 7. Samples of vessel and background likelihood PDFs. Vessel likelihood PDFs (not normalized) $p_v(y_t|r_t = r_t^i)$, for different radius values r_t^i, are extracted from the joint density depicted in Fig. 5. Background likelihoods $p_{bg}(y_t)$ are independent of the model, but are specific to a given dataset.

4.3. *Vessel model transition prior* $p(x_t|x_{t-1})$

The first order Markovian transition *prior* $p(x_t|x_{t-1})$ in Eq. (7) corresponds to prior knowledge on the potential variations of the geometric model. We chose to express the transition prior as a function of radius and direction variations, with $x_t = (p_t, r_t, d_t)$:

$$p(x_t|x_{t-1}) = p(p_t, r_t, d_t|p_{t-1}, r_{t-1}, d_{t-1}) \\ \propto p(r_t|r_{t-1})p(d_t|d_{t-1}, r_{t-1}) \qquad (13)$$

We assume here that the prior does not depend on the centerline position p_t (*i.e.* spatial invariance), so $p(p_t, r_t, d_t|p_{t-1}, r_{t-1}, d_{t-1}) = p(r_t, d_t|r_{t-1}, d_{t-1})$, and that scale variations are independent of the direction (*i.e.* vessel orientation invariance), keeping only two terms, the scale transition prior $p(r_t|r_{t-1})$ and the direction prior $p(d_t|d_{t-1}, r_{t-1})$. This model is very natural and transitions in radius and scale can be learned from an annotated database.

4.3.1. *Scale transition prior* $p(r_t|r_{t-1})$

The scale transition PDFs $p(r_t|r_{t-1} = r_{t-1}^i)$, depending on the realization r_{t-1}^i (previous radius), are extracted from the joint density $\hat{f}(r_t, r_{t-1})$, which is learned from successive radius samples (r_t^i, r_{t-1}^i) from the ground-truth-fitted models as illustrated in Fig. 8. It can

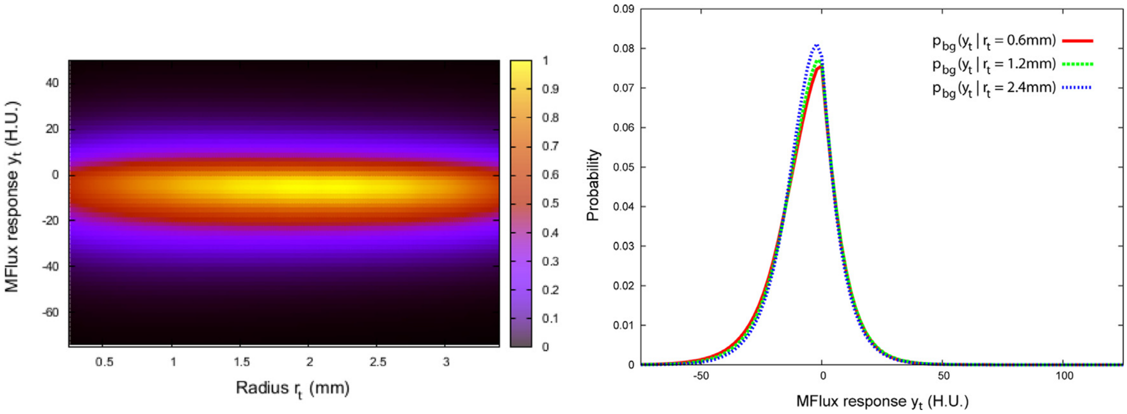

Fig. 6. Background likelihood. Left: for a selected dataset, joint density $\hat{f}_{bg}(y_t, r_t)$ estimated in the background ($h_{y_t} =$ 10 H.U. and $h_{r_t} = 0.3$ mm). Right: background likelihood PDFs (not normalized) $p_{bg}(y_t|r_t = r_t^i)$ for different radiuses r_t^i.

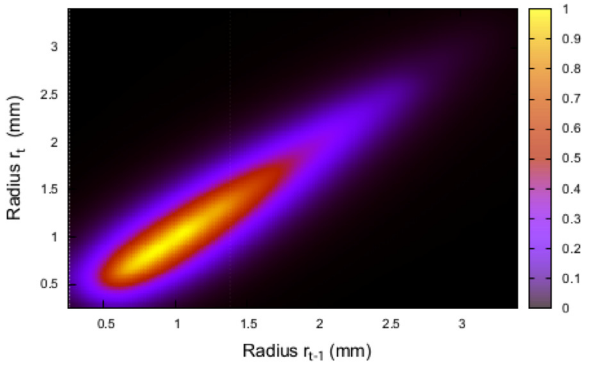

Fig. 8. Joint density $\hat{f}(r_t, r_{t-1})$ of successive radiuses, obtained from samples (r_t^i, r_{t-1}^i), using kernel estimation with bandwidth parameters $h_{r_t} = h_{r_{t-1}} = 0.3$mm, and a discretization step $s = 0.3$ mm. Learning was performed on 10 datasets randomly selected in our database.

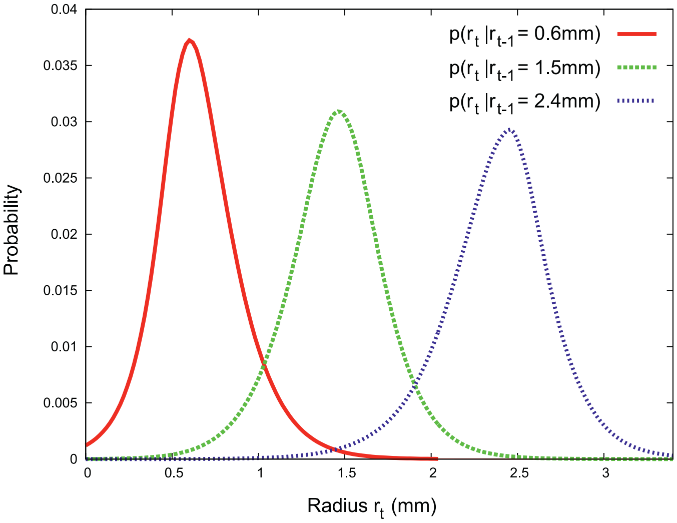

Fig. 9. Samples of scale transition priors. PDFs of scale transition priors $p(r_t|r_{t-1} = r_{t-1}^i)$, for different radius values r_{t-1}^i, were extracted from the joint density depicted in Fig. 8.

be observed that areas of higher density lie along the identity line $r_t = r_{t-1}$, confirming the intuition that scale variations along coronary arteries are relatively slow and that an artery retains a locally stable radius. In Fig. 9, we provide samples of transition priors $p(r_t|r_{t-1} = r_{t-1}^i)$ for selected radius realizations r_{t-1}^i, highlighting increased variance for larger vessels. In general, these distributions are not symmetric and not exactly centered around the previous scale r_{t-1}^i, but rather at a slightly lower value, related to the fact that coronary arteries tend to decrease in radius from the ostia to distal ends, which corresponds to the order of our tracking scheme. This observation holds mainly for "larger", clinically relevant vessels (diameter ≥ 1.5 mm). For smaller vessels (e.g. 0.5 mm), one admittedly reaches accuracy limits for the original "ground truth" segmentation, which makes interpretations delicate. By highlighting the asymmetry, we show that our non-parametric model is able to capture relatively subtle trends in the training data.

4.3.2. Direction prior $p(d_t|d_{t-1}, r_{t-1})$

Blood vessels generally follow relatively smooth trajectories. The directional prior term $p(d_t|d_{t-1}, r_{t-1})$ is designed to embed such knowledge, defined through angular variations. We decompose d_t as follows:

$$d_t = \cos(\alpha_t) d_{t-1} + \sin(\alpha_t)\big(\cos(\beta_t)v_1 + \sin(\beta_t)v_2\big) \quad (14)$$

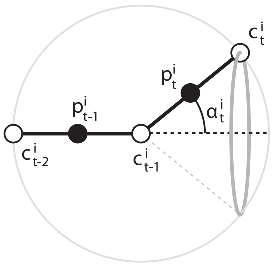

Fig. 10. Direction prior and tangential angle α_t. The angle α_t is defined as the angular deviation between two consecutive directions d_{t-1} and d_t.

where (d_{t-1}, v_1, v_2) forms an orthonormal basis, $\alpha_t = \arccos d_{t-1} \cdot d_t$ is the tangential angle enclosed between d_{t-1} and d_t (see Fig. 10) and β_t is the second rotation angle. For N_t discrete samples, we can express the direction prior density as follows [54]:

$$p(d_t|d_{t-1}, r_{t-1}) = \sum_{i=1}^{N_t} \tilde{w}^i \delta(d_t - d_t^i) \quad (15)$$

where $\delta(.)$ is the Dirac delta function and $\tilde{w}^i$ are normalized weights that we define as:

$$\tilde{w}^i = \frac{p(\alpha_t^i|r_{t-1})}{\sum_{k=1}^{N_t} p(\alpha_t^k|r_{t-1})} \quad (16)$$

The direction prior is thus modeled as simply proportional to the distribution of the tangential angle $p(\alpha_t^i|r_{t-1})$. For the sake of simplicity, we assume that the second angle β_t is uniformly distributed.

In spirit, our approach is similar to the ones in [54] and [1], where fixed parametric distribution models (respectively Gaussian and von-Mises–Fisher) were used. In contrast, we again learn the directional PDFs in a non-parametric fashion. Fig. 11 illustrates the joint density $\hat{f}(\alpha_t, r_{t-1})$ estimated from our database, along with examples of densities $p(\alpha_t|r_{t-1} = r_{t-1}^i)$ for different radius values. Note that smaller vessels exhibit higher angular variations, and that the maximal probability density is attained for a small, but non-zero angle.

5. Particle filtering for vessel tracking

Particle filtering techniques, also known as Sequential Monte-Carlo methods, rely on the stochastic evolution of a population of samples (the so-called particles) for the estimation of non-linear, non-Gaussian Bayesian processes [2,14]. Tracking with particle filtering consists in recursively estimating the posterior distribution of the vessel model $p(x_{0:t}|z_{1:t})$. To the authors' knowledge, particle filters were first used for 3D vascular segmentation in [16,17]. This seminal work inspired several recent developments [1,42,49,52,54,55,62–64,68]. These techniques differ in their applicative scopes, their transition and observation models, their sampling schemes and proposed refinements. These different works have demonstrated the attractiveness of this generic, highly flexible framework for vascular segmentation, in particular in terms of robustness to local perturbations (pathologies and image artifacts). In this work, we use particle filtering to optimize the Bayesian model discussed in Section 4. In the following, we briefly describe the core principles of particle filtering, its key issues and the existing refinements related to our approach.

5.1. Monte-Carlo estimation

Assuming the possibility of simulating N random samples $\{x_{0:t}^i\}_{i=1,\dots,N}$ identically distributed from the posterior distribution $p(x_{0:t}|z_{1:t})$ (*i.e. perfect Monte-Carlo sampling*), we obtain an empirical estimate $\hat{p}(x_{0:t}|z_{1:t})$ of $p(x_{0:t}|z_{1:t})$ as:

$$\hat{p}(x_{0:t}|z_{1:t}) = \frac{1}{N}\sum_{i=1}^{N} \delta(x_{0:t} - x_{0:t}^i)$$

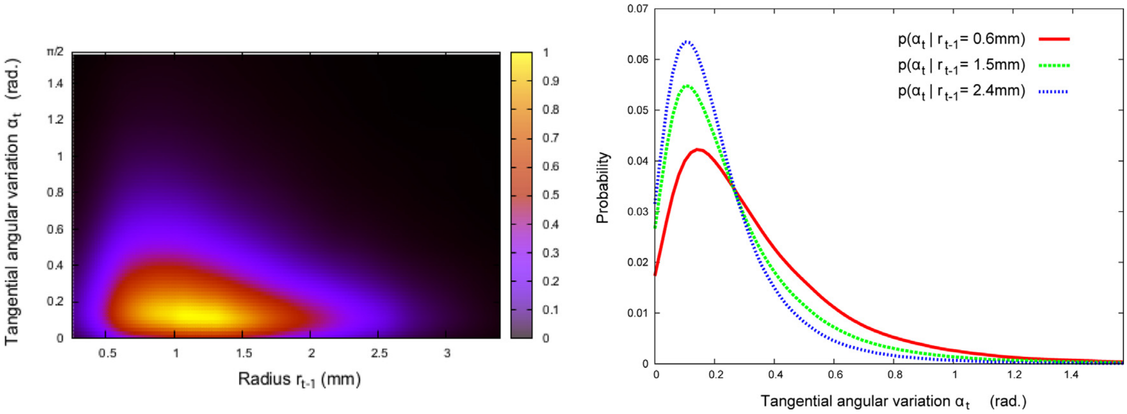

Fig. 11. Estimation of the direction prior. Left: joint density $\hat{f}(\alpha_t, r_{t-1})$ of angular variations and radiuses. The density was obtained from samples (α_t^i, r_{t-1}^i), using kernel estimation with bandwidth parameters $h_{\alpha_t} = 0.05$ rad and $h_{r_{t-1}} = 0.3$ mm and a discretization step $s = 0.3$ mm. Right: samples of PDFs $p(\alpha_t | r_{t-1} = r_{t-1}^i)$, for different radius values r_{t-1}^i, extracted from the joint density. Learning was performed on 10 datasets randomly selected in our database.

where $\delta(.)$ is the Dirac measure. In other words, the posterior distribution is approximated by a discrete sum of point-wise estimates at sampling locations. These samples $x_{0:t}^i$ are also referred to as the *particles*.

Importance sampling relies on the use of a so-called *importance sampling distribution* (or proposal distribution) $q(x_{0:\,t}|z_{1:\,t})$, to circumvent the issue of not being able to sample from the posterior distribution $p(x_{0:\,t}|z_{1:\,t})$ [24]. The core idea of importance sampling is to obtain samples from $q(x_{0:\,t}|z_{1:\,t})$ and *correct* them with respect to $p(x_{0:\,t}|z_{1:\,t})$ through the introduction of *importance weighting*:

$$w(x_{0:t}) \propto \frac{p(x_{0:t}|z_{1:t})}{q(x_{0:t}|z_{1:t})} \tag{17}$$

From N particles $\{x_{0:t}^i\}_{i=1,\dots,N}$ identically distributed from $q(x_{0:\,t}|z_{1:\,t})$, a valid Monte-Carlo approximation of the posterior $p(x_{0:\,t}|z_{1:\,t})$ is now given by:

$$\hat{p}(x_{0:t}|z_{1:t}) = \sum_{i=1}^{N} \tilde{w}_t^i \delta(x_{0:t} - x_{0:t}^i) \tag{18}$$

where $\{\tilde{w}_t^i\}$ are the normalized importance weights associated with the particles:

$$\tilde{w}_t^i = \frac{w(x_{0:t}^i)}{\sum_{j=1}^{N} w(x_{0:t}^j)}.$$

Sequential importance sampling (SIS) consists in using importance sampling in a sequential, recursive fashion, without modifying past samples. The aim is to keep the complexity of the sequential estimation process from increasing with the temporal length of the sequence. The key to achieve that goal is to consider an importance distribution that factorizes as follows:

$$q(x_{0:t}|z_{1:t}) = q(x_t|x_{0:t-1}, z_{1:t})q(x_{0:t-1}|z_{1:t-1}) \tag{19}$$

This allows the recursive evaluation of the importance weights [14]:

$$w_t^i \propto w_{t-1}^i \frac{p(z_t|x_t^i)p(x_t^i|x_{t-1}^i)}{q(x_t^i|x_{0:t-1}^i, z_{1:t})}.$$

Coupling the SIS algorithm with a particle resampling step, one obtains the generic *Sampling-Importance-Resampling* (SIR) algorithm, also known as Bootstrap [27] or Condensation filter [34].

5.2. Key issues and existing refinements

5.2.1. Choice of importance distribution

A key design issue for particle filters is the choice of an adequate importance distribution, This choice can be intricate in practice and directly influences the robustness and computational efficiency of the overall tracking [2,15].

In general, one will employ a (prediction) importance distribution which depends only on x_{t-1} and possibly on z_t, so that $q(x_t|x_{0:t-1}^i, z_{1:t}) = q(x_t|x_{t-1}^i, z_t)$. A particularly popular choice of importance distribution is the use of the transition prior [27,34]. Setting $q(x_t|x_{t-1}^i, z_t) = p(x_t|x_{t-1}^i)$, the weight update simplifies to an iterative correction through the likelihood: $w_t^i \propto w_{t-1}^i p(z_t|x_t^i)$. Despite its simplicity, this importance distribution may exhibit degraded performance in case of strong disagreement between the prior and the likelihood, as it does not take into account the latest observations z_t.

In a previous work [42], we showed how one can build a discrete approximation of the posterior distribution while taking into account the latest available observations. We demonstrated that sampling from such an importance distribution considerably reduces the number of samples needed to obtain good tracking performance. This approach suffers from two main drawbacks. First, the discretization of the search space negatively impacts the estimation accuracy. Second, the construction of the full posterior can quickly become prohibitively costly for complex models.

Among general refinements aiming at designing better importance distributions, one can cite Rao-Blackwellization [7], the *unscented transform* [48], bridging densities [25] and progressive correction schemes [50]. Another popular approach is the family of *auxiliary* particle filters [51], which we employ in this work and discuss in Section 6.

5.2.2. Adaptation of the number of particles

The number of particles required to obtain "good" estimates depends notably on the complexity and dimensionality of the problem at hand. In a vast majority of works, N is kept fixed and is set empirically as a tradeoff between robustness, accuracy and computational cost.

The number of particles can also be adapted *dynamically*, following an iteration-dependent number N_t. The goal is generally twofold: increase the quality of the estimates (more samples in complex situations) and improve computational efficiency (less samples in easy situations). Such schemes can be found in the works of [18,40,57], relying on Kullback–Leibler and entropy measures, respectively, to evaluate the degeneracy of the filter and adapt the number of particles accordingly. In this work, we reuse the Effective Sample Size (ESS) approach of [58], as discussed further in Section 6.

6. Adaptive Auxiliary Particle Filtering (AAPF)

Our approach, which we refer to as *Adaptive Auxiliary Particle Filtering (AAPF)*, combines two refinements of particle filters: (1) an adaptive particle sampling scheme, known as *auxiliary* sampling [51], to construct the importance distribution, (2) a dynamic allocation of the number of particles with respect to the complexity of the local estimation of the underlying vessel model.

6.1. Adaptation of the importance distribution: Auxiliary Particle Filters

The principle of *auxiliary* particle filtering (APF) [51] is to construct the importance distribution of the sampled particles empirically, via simulation, to take into account the current data observations z_t. The particle filter is said to be *adapted*, as the auxiliary sampling is a two-pass process, where the first *simulation* pass allows a one-step look-ahead to improve the effective allocation of the particles in the second pass.

We refer the reader to [51] for in-depth discussions of the theoretical background behind Auxiliary Particle Filters. In a nutshell, the APF algorithm can be described in three main steps:

1. construction of an empirical importance distribution: sampling of an auxiliary population $\{\mu_t^i\}$ and computation of associated weights $\{\omega_t^i\}$;
2. resampling from $\{\mu_t^i, \omega_t^i\}$ to select the indices $\{k^i\}$, of the particles $\{x_{t-1}^{k^i}\}$ at time $t-1$;
3. sampling of the new population $\{x_t^i, \tilde{w}_t^i\}$: prediction of new states $\{x_t^i \sim p(x_t|x_{t-1}^{k^i})\}$, and update of the weights $\{\tilde{w}_t^i\}$.

In effect, the auxiliary population $\{\mu_t^i, \omega_t^i\}$ can be seen as a one-step look-ahead. It is used to select indices $\{k^i\}$ of the particles at time $t-1$ in an informative fashion. A key design choice is thus how one obtains the auxiliary population $\{\mu_t^i, \omega_t^i\}$. In this work, we simply use samples from the transition prior, with $\mu_t^i \sim p(x_t|x_{t-1}^i)$. This approach is similar to [17] and [57], where the authors also implemented this particular case of APF and demonstrated its efficiency. The weights $\{\omega_t^i\}$ for the auxiliary population are simply given by

$$\omega_t^i = p(z_t|\mu_t^i)\tilde{w}_{t-1}^i$$

where $p(z_t|\mu_t^i)$ is the likelihood for the auxiliary particle μ_t^i and $\tilde{w}_{t-1}^i$ is the weight of the particle x_{t-1}^i that spawned μ_t^i.

From the auxiliary population $\{\mu_t^i, \omega_t^i\}$, classical resampling is employed to obtain the indices $\{k^i\}$ at time $t-1$. We thus obtain a population of particles $\{x_{t-1}^{k^i}\}$, which is hopefully better distributed with regards to the "future" iteration at time t. This population is then evolved in a classical fashion through the transition prior:

$$x_t^i \sim p(x_t|x_{t-1}^{k^i})$$

A last key element of the APF algorithm is the update of the weights, to obtain the final weighted population $\{x_t^i, w_t^i\}$. In our special case using the transition prior for simulation, the weight update is simply the following likelihood ratio:

$$w_t^i = \frac{p(z_t|x_t^i)}{p(z_t|\mu_t^{k^i})} \quad (20)$$

We refer the reader to [17,51,57] for detailed mathematical derivations.

If μ_t^i is obtained as a sample from $p(x_t|x_{t-1}^i)$ as in our case, one iteration of APF corresponds to two consecutives SIS passes with intermediate resampling. Another interpretation is that APF first performs a selection with replacement on the population at $t-1$, with respect to an empirical approximation of $p(x_{t-1}|z_{1:t})$ (one-step look-ahead). It then evolves the population classically through the transition prior $p(x_t|x_{t-1})$.

Providing that the point-wise estimates $\{\mu_t^i\}$ give a good characterization of $p(x_t|x_{t-1}^i)$, samples generated by the APF are likely to be accurately located close to the true state x_t. Compared to classical SIS/SIR using the priors as importance distribution, the two-pass design of the APF generally reduces degeneracy issues, decreasing the variance of the weights by incorporating information on the most recent observations. This is obtained with higher computational complexity as it actually requires the evaluation of 2N particles per iteration.

6.2. Dynamic adaptation of the number of particles

As was previously mentioned, a possible refinement of standard particle filtering is to vary *dynamically* the number of samples. We now consider a varying number N_t of particles to be determined at each iteration. Among the existing techniques cited in Section 5.2, we reused the idea of [58] for its simplicity and intuitiveness.

The authors in [58] proposed to set N_t so as to attain a *fixed* effective sample size (ESS) N^*. The ESS was first introduced in [3] and [47] as a measure of the degree of degeneracy affecting the filter. In order to attain N^*, a relatively low number of particles N_t will be required in situations where the importance distribution closely coincides with the filtered posterior (*i.e.* low variance of the weights). Conversely, a larger N_t will be needed in more difficult cases, such as noisy, low-contrast vessels, bifurcations and pathological cases.

The ESS of a population of N_t particles is defined as:

$$ESS(N_t) = \frac{N_t}{1 + d(p,q)} \quad (21)$$

Where $d(p, q)$ is the χ^2 distance between the marginal posterior $p(x_t|z_{1:t})$ and the importance distribution $q(x_t|x_{t-1}, z_t)$. Fixing a target ESS N^*, we have:

$$N_t = \lceil N^*(1 + d(p,q)) \rceil \quad (22)$$

The χ^2 distance $d(p, q)$ is not available analytically and has to be integrated numerically. Assuming the availability of an initial population $\{x_t^i, \tilde{w}_t^i\}_{i=1,\ldots,N_t^1}$ of N_t^1 samples from $q(x_t|x_{t-1}, z_t)$ with associated (normalized) importance weights, $d(p, q)$ can be estimated empirically by the coefficient of variation cv^2:

$$\hat{d}(p,q) = cv^2 = \frac{N_t^1}{\sum_{i=1}^{N_t^1}(\tilde{w}_t^i)^2} - 1 \quad (23)$$

By combining Eqs. (23) and (22), we can derive the number of particles N_t as:

$$N_t = \left\lceil N^* \left(\frac{N_t^1}{\sum_{i=1}^{N_t^1}(\tilde{w}_t^i)^2} \right) \right\rceil \quad (24)$$

One always has $N_t \geq N^*$. We chose to rely on the estimation of $d(p, q)$ by cv^2 for the sake of simplicity and computational efficiency. The major issue with this scheme is the need for an initial population of weighted samples $\{x_t^i, \tilde{w}_t^i\}_{i=1,\ldots,N_t^1}$. In our work, we propose to combine auxiliary particle filtering (APF) with the use of an adaptive number of particles, using the simulation phase of the APF to estimate N_t.

6.3. Adaptive Auxiliary Particle Filtering (AAPF)

We now show how both APF and dynamic adaptation of the number of particles can be combined in a natural fashion, yielding our adaptive auxiliary particle filtering algorithm (AAPF).

The estimation of the number of particles N_t requires an initial population of N_t^1 samples $\{x_t^i, \tilde{w}_t^i\}_{i=1,\ldots,N_t^1}$. We propose to use the simulated population $\{\mu_t^i, \omega_t^i\}_{i=1,\ldots,N_{t-1}}$ of the APF to evaluate N_t. The pseudo-code description of our algorithm is provided in Fig. 12. The main difference with classical APF is the estimation of N_t at line 8. We discuss the choice of target ESS N^* in Section 8 with respect to our application. This parameter intuitively balances overall estimation quality with computational cost.

From a theoretical point of view, it is important to note that the initial population $\{\mu_t^i, \omega_t^i\}_{i=1,\ldots,N_{t-1}}$ on which N_t is estimated does not

```
AAPF($\{x^i_{t-1}, \tilde{w}^i_{t-1}\}_{i=1,\dots,N_{t-1}}$)
       ▷ Simulation of auxiliary population $\{\mu^i_t, omega^i_t\}$
 1     for $i \leftarrow 1$ to $N_{t-1}$
 2         do
 3             $\mu^i_t \sim p(x_t|x^i_{t-1})$
 4             $\omega^i_t \leftarrow p(z_t|\mu^i_t)\tilde{w}^i_{t-1}$
       ▷ Normalization
 5     for $i \leftarrow 1$ to $N_{t-1}$
 6         do
 7             $\omega^i_t \leftarrow \frac{\omega^i_t}{\sum_{j=1}^{N_{t-1}} \omega^j_t}$
       ▷ Estimation of the number of particles $N_t$ (Eq. 24)
 8     $N_t \leftarrow \lceil N^*(\frac{N_{t-1}}{(\sum_{i=1}^{N_{t-1}} (\omega^i_t)^2)})\rceil$
       ▷ Sampling of the indices at time $t-1$
 9     $\{k^i\}_{i=1,\dots,N_t} \leftarrow$ RESAMPLE($\{\mu^i_t, \omega^i_t\}_{i=1,\dots,N_{t-1}}$)
       ▷ Sampling of the new population at time $t$
10     for $i \leftarrow 1$ to $N_t$
11         do
12             $x^i_t \sim p(x_t|x^{k^i}_{t-1})$
13             $w^i_t \leftarrow \frac{p(z_t|x^i_t)}{p(z_k|\mu^{k^i}_t)}$ ▷ Eq. 20
       ▷ Normalization
14     for $i \leftarrow 1$ to $N_t$
15         do
16             $\tilde{w}^i_t \leftarrow \frac{w^i_t}{\sum_{j=1}^{N_t} w^j_t}$
```

Fig. 12. Auxiliary particle filter with adaptive number of particles (AAPF). Obtention of the weighted particle population $\{x^i_t, \tilde{w}^i_t\}_{i=1,\dots,N_t}$ from $\{x^i_{t-1}, \tilde{w}^i_{t-1}\}_{i=1,\dots,N_{t-1}}$ (single iteration).

come from the true APF importance distribution [51], but from the transition prior used for the simulation (line 3). This means that the coefficient of variation of these initial weights is not necessarily an accurate estimation of $d(p, q)$. It is, however, a good measure of the complexity of the current iteration. If the simulation by the transition prior leads to a large variance of these weights, chances are that the APF will also have a relatively large weight variance. In practice, the APF generally reduces the weight variance by better allocating subsequent samples, so that this scheme tends to slightly overestimate N_t. In difficult situations, our technique reinforces the correction from APF by allocating more samples. As demonstrated in Section 8, this combination leads to a quantitative gain in robustness. In particular, it results in a more robust behavior at bifurcations, easing their explicit detection (Section 7.3).

7. AAPF for CTA coronary segmentation

The AAPF algorithm described in the previous section is generic and could be applied to any Bayesian sequential estimation problem. We now discuss its particular use for the segmentation of coronary vessels on CTA data with our dedicated vessel model (Section 4). We recall that our state-space model is defined as $x_t = (p_t, d_t, r_t)$, with p_t the centerline position, d_t the tangential position and r_t the vessel radius. Particles $\{x^i_t\}$ evolved by the AAPF are thus state realizations of our model.

The workflow employed for our particle filtering approach is as follows:

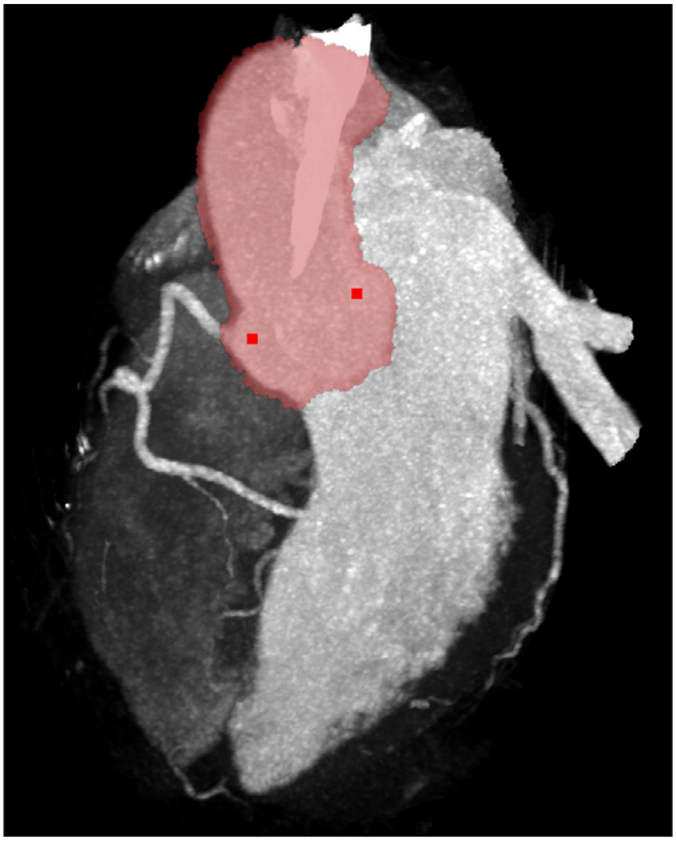

Fig. 13. Sample result of automatic aorta segmentation and ostia detection. On this 3D maximal intensity projection view of a cardiac CTA dataset, the segmented aorta is highlighted, with the two (left and right) detected ostia points plotted as red squares. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

- automatic aorta segmentation is obtained through a fast variant [28] of the isoperimetric graph-based algorithm [29];
- ostia points (location of the coronary arteries branching off the aorta) are detected automatically following the principles used in [13,30,59];
- from each of the detected ostia locations, we initialize our AAPF filter with a population of N_0 particles sampled uniformly among all possible directions and radiuses. We used $N_0 = 2000$ in our tests.

The first two steps provide the automatic initialization of our algorithm (see Fig. 13).

The AAPF algorithm involves the evaluation of likelihood terms (see Section 4.2) and sampling from transition priors (see Section 4.3). The different distributions were all learned in a non-parametric fashion from our ground-truth database (see Section 4.1). The learning process is dependent on the discretization resolution of the geometric model. We used a fixed step $s = 0.3$ mm, of the order of the intra-slice resolution of our CTA data. This discretization step corresponds to the spacing between control points $\|c_{t-1} - c_t\| = s, \forall t$, and controls the spatial progression of the particle filter along the vessels.

In the following, we first discuss the evaluation of the likelihood terms (Section 7.1) and the sampling from the prior distributions (Section 7.2). As the AAPF propagates along the coronary tree, we use mean-shift clustering to explicitly detect bifurcations (Section 7.3). Each branch is evolved independently, until a stopping criterion is fulfilled (Section 7.4). A compact centerline+radius representation of the extracted tree is obtained based on the mean-shift modes (Section 7.5).

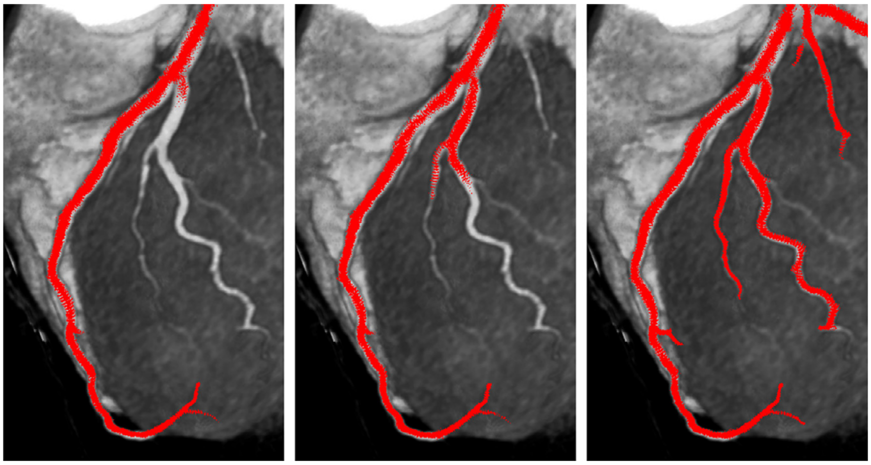

Fig. 14. Effect of adaptive particle filtering and mean-shift detection. Left: classical SIR filter. Middle: our implementation of auxiliary particle filter with adaptive number of particles (AAPF). Right: AAPF with mean-shift detection of bifurcations and independent tracking of multiple branches. Successive particle populations (in red) are overlayed on a volume rendering of cardiac CTA data. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

Table 1
Parameters used for the discretization of the different distributions of our model.

Variable	Discretiz. step	Range	# of values
Radius values: r_{t-1}, r_t	0.03 mm	[0.1, 3.97]	$N_r = 130$
`MFlux` responses: y_t	0.5 H.U.	[−49.5, 150]	$N_y = 400$
Angular variation: α_t	$\frac{\pi}{198}$ rad	$[0, \frac{\pi}{2}]$	$N_\alpha = 100$

7.1. Evaluation of the likelihood terms

We recall from Eq. (12) that our likelihood model is decomposed as:

$$p(z_t|x_t = x_t^i) = \frac{p_v(y_t^i|r_t = r_t^i)}{p_{bg}(y_t^i)} \prod_{y_t^j \in z_t, j=1,\ldots,N_t} p_{bg}(y_t^j) \tag{25}$$

where observations at time t are $z_t = \{y_t^i\}_{i=1,\ldots,N_t}$, with y_t^i an elementary observation obtained as the `MFlux` feature response for the state realization (particle) $x_t^i = (p_t^i, d_t^i, r_t^i)$. Terms independent of the realization, such as the product of background likelihoods $\prod_{y_t^j \in z_t, j=1,\ldots,N_t} p_{bg}(y_t^j)$, can be dropped.

Vessel and background distributions p_v and p_{bg} were learned from non-parametric joint densities (see Section 4.1). We store them as histogram densities, discretized at relatively fine resolutions and directly integrated from the learned joint densities. Chosen discretization parameters (Table 1) correspond to tradeoffs between memory requirements and potential undesirable discretization effects. In practice, the histograms are used as simple lookup tables to value the likelihood terms. To avoid numerical instabilities in likelihood ratios $\frac{p_v(y_t^i|r_t=r_t^i)}{p_{bg}(y_t^i)}$, we fix a lower bound value $p_{min} = 10^{-6}$ on $p_{bg}(y_t^i)$.

7.2. Sampling from the prior densities

Besides likelihood evaluation, AAPF requires the ability to sample from priors $p(x_t|x_{t-1} = x_{t-1}^i)$. Our transition model (Section 4.3) is given by:

$$p(x_t|x_{t-1}^i) = p(r_t|r_{t-1}^i)\,p(d_t|d_{t-1}^i, r_{t-1}^i) \tag{26}$$

The radius and directional transition priors were also learned non-parametrically. To be able to sample from these distributions, we first discretize them, similarly to the likelihood terms, and store them as histograms. From these histogram representations, we obtain new radiuses and angular variations using multinomial sampling.

For a given particle $x_{t-1}^i = (p_{t-1}^i, d_{t-1}^i, r_{t-1}^i)$, we obtain $x_t^i = (p_t^i, d_t^i, r_t^i)$ as follows:

1. sample a new radius $r_t^i \sim \mathcal{H}(r_t|r_{t-1}^i)$ via multinomial sampling from the radius prior histogram for r_{t-1}^i;
2. sample a new direction d_t^i:
   - construct an orthonormal basis (d_{t-1}^i, v_1, v_2) (d_{t-1}^i set as the x-axis by convention);
   - sample angle $\alpha_t^i \sim \mathcal{H}(\alpha_t|r_{t-1}^i)$ via multinomial sampling from the angular prior histogram for r_{t-1}^i (see Section 4.3.2);
   - sample angle $\beta_t^i \sim \mathbb{U}[0, 2\pi[$ (second rotation angle sampled uniformly);
   - obtain $d_t^i = cos(\alpha_t^i)d_{t-1}^i + sin(\alpha_t^i)(cos(\beta_t^i)v_1 + sin(\beta_t^i)v_2)$ (combination of the rotations);
3. compute the new position $p_t^i = p_{t-1}^i + \frac{s(d_t^i + d_{t-1}^i)}{2}$ (with $s = 0.3$mm for the spatial discretization step).

7.3. Detection of bifurcations by mean-shift clustering

With respect to the posterior distribution, bifurcations correspond to the emergence of multiple modes. Although particle filters are theoretically able to cope with multi-modal distributions, they raise practical issues in sequential estimation schemes. Because of the sample impoverishment induced by the resampling procedures, secondary modes are quickly lost as all particles are re-allocated to the dominant mode. In our applicative context, this means that the filter will robustly track only one branch. As can be seen in Fig. 14 (left and middle), secondary branches are progressively depopulated and lost. Compared to classical SIR implementations, our AAPF algorithm improves the behavior at bifurcations. As illustrated in Fig. 14, AAPF generally tracks secondary branches on a longer portion than SIR. More importantly, small, secondary branches are frequently missed by the SIR implementation. AAPF, on the other hand, consistently achieves a better sampling of asymmetric bifurcations. We explain this observation by a combination of auxiliary particle filtering and the dynamic adaptation of the number of samples. The number of particles selected by our scheme tends to be larger in bifurcation areas, as the emergence of multiple modes tends to increase the variance of the weights (see Fig. 15).

Although AAPF sensibly improves the algorithm's robustness at bifurcations, it is not able to track several branches in parallel until their distal ends. Secondary branches are still lost after a while (Fig. 14, middle). In particular, one can note in Fig. 15 (right) that following a peak at bifurcations, the number of allocated particles tends

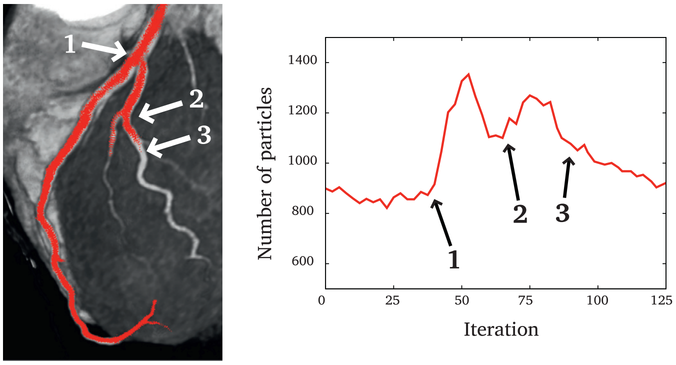

Fig. 15. Adaptive number of particles and behavior at bifurcations. Left: successive populations of particles from an AAPF propagation (see also Fig. 14). Right: variation of the number N_t of allocated particles (for $N^* = 500$) per iteration. One can notice an increase of N_t at bifurcations (arrows 1 and 2). This number progressively decreases (arrow 3) as secondary branches are lost because of sample impoverishment.

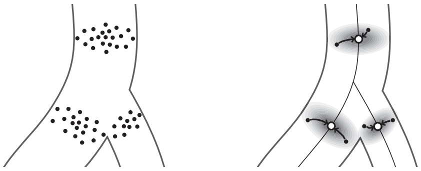

Fig. 16. Mean-shift clustering for branching detection and centerline estimation. Left: at bifurcations, the population of particles (weighted spatial positions $\{p_t^i, \tilde{w}_t^i\}$) splits into different clusters. Right: mean-shift clustering operates, for each initial position p_t^i, as a hill-climbing procedure towards local maxima of density of $\hat{f}_h(p_t|z_{1:t})$. The detected modes $\{\hat{m}_t^j\}_{t=0,\dots,L}$ (white circles) are used as an estimation of the centerline position.

to decrease as more and more particles are re-allocated to the dominant mode. To cope with the progressive loss of secondary branches, several works have proposed schemes to explicitly detect bifurcations. [1,16,17] used K-means clustering of samples. [20] used spectral clustering on non-stochastic candidates. [49] used quality threshold (QT) clustering of samples. In this work, we prefer mean-shift mode detection [21], motivated from both a theoretical and a practical point of view. First, mean-shift procedures can be interpreted in terms of kernel density estimation, which integrates seamlessly with Monte-Carlo techniques. This allows us to preserve a homogeneous theoretical framework. Second, mean-shift clustering does not require specific initialization or the specification of the number of clusters. In our method, mean-shift serves two purposes: we use it for bifurcation detection and for the extraction of the final result (see Section 7.5).

To detect coronary bifurcations, we apply mean-shift clustering on the particle positions with associated importance weights $\{p_t^i, \tilde{w}_t^i\}_{i=1,\dots,N_t}$ (see Fig. 16). This corresponds to a marginalization of our state space $x_t = (p_t, d_t, r_t)$. We recall that the Monte-Carlo importance estimation of the position posterior $p(p_t|z_{1:\,t})$ is given by $\hat{p}(p_t|z_{1:t}) = \sum_{i=1}^{N_t} \tilde{w}_t^i \delta(p_t - p_t^i)$. Our mean-shift procedure works on a continuous density estimated from $\{p_t^i, \tilde{w}_t^i\}_{i=1,\dots,N_t}$:

$$\hat{f}_h(p_t|z_{1:t}) = \sum_{i=1}^{N_t} \tilde{w}_t^i K_H(p_t - p_t^i)$$

This density is closely related to the Monte-Carlo approximation of the posterior density. A key difference is the regularization induced by the mean-shift kernel.

7.3.1. *Kernel and bandwidth selection*

In this work, we use the Epanechnikov kernel for its simplicity and computational efficiency. The shadow (derivative) of this kernel is uniform, so that mean-shift computations reduce to simple weighted means.

A key parameter is the bandwidth parameter h, *i.e.*, the radius of the Epanechnikov shadow. Automatic bandwidth selection is an intricate issue [8,10]. Fortunately, we can leverage our application context to alleviate this issue. As our goal is to detect separated spatial clusters (different branches), we simply set the bandwidth h_t to the Monte-Carlo estimate of the radius r_t at iteration t:

$$h_t = \hat{\mathbb{E}}[p(r_t|z_{1:t})] = \sum_{i=1}^{N_t} w_t^i r_t^i \tag{27}$$

The bandwidth value is then adapted at each iteration. The Monte-Carlo estimate of the radius is mainly driven by the dominant branch, so that the scale of the mean-shift kernel is adapted to the current (main) vessel of interest. We found this heuristic choice to provide a good tradeoff between false positives (spurious modes detected because of a too low value of h) and the early detection of true bifurcations.

7.3.2. *Computational efficiency*

A counterpart to the robustness of mean-shift clustering is its relatively high computational cost (complexity of $O(N_t^2)$). Approximated variants [9,61] have been proposed to accelerate its computation. For the sake of simplicity, we limit ourselves to a textbook implementation of mean-shift clustering, but still introduce refinements to limit its computational impact on the overall algorithm.

As bifurcations are relatively rare occurrences, the tracked density exhibits a single mode in a vast majority of cases. At each iteration of our filter, we propose to probe the density as follows:

- we first detect an initial mode m_t^1, by applying mean-shift to the particle x_t^i of highest weight $\tilde{w}_t^i$;
- we then sample a small sub-population of test particles $\{x_t^i\}_{i=1,\dots,N_p}$, with $N_p < N_t$. If at least one new mode is discovered $(m_t^2 \neq m_t^1)$, full mean-shift clustering is performed.

For the detection of the first mode, note that one could actually select any particle. Our scheme generally accelerates the procedure by selecting a particle likely to lie close to a local maximum of density. Secondary test particles are sampled according to $\|p_t^i - m_t^1\|$, their distance to the first mode. Intuitively, the farthest a particle is from the first mode, the more likely it belongs to another (hypothetical) mode.

On our cardiac CTA data, actual bifurcations are present in less than 3% of the filter iterations on average. In our tests, fixing N_p as low as 5% of N_t was enough to robustly detect the presence of all actual bifurcations. This means that in 97% of cases, our bifurcation detection scheme corresponds to applying mean-shift on only 5% of the particle population. This simple, empirical approach alleviates the burden of mean-shift clustering, keeping our overall algorithm at good levels of computational efficiency (see Section 8).

When a bifurcation is assessed (with two, occasionally three modes detected), propagation is resumed independently for each cluster. The re-population of each cluster is ensured naturally by the AAPF (line 9 in Fig. 12). Additionally, we update an inclusion mask with each newly extracted branch. Checking for inclusion in this mask avoids retracing the same branch twice.

7.4. *Stopping criterion*

We implemented a stopping criterion based on the likelihood ratios $\frac{p_v(y_t^i|r_t=r_t^i)}{p_{bg}(y_t^i)}$. We recall that a particle x_t^i whose likelihood ratio is lower than 1 is considered as more likely to correspond to the background than to an actual vessel.

For increased robustness, we base our analysis on the entire population $\{x_t^i\}_{i=1,\dots,N_t}$. If less than F% of the particles at iteration t have likelihood ratios lower than 1, the iteration is flagged. Propagation is stopped if more than half of the last S iterations are flagged. We empirically set $F = 25\%$ and $S = 20$ in our implementation. Results showed relatively low dependence on F in our experiments. The parameter S controls the tolerance to local anomalies. It allows the filter to locally loose the target vessel, wander and potentially re-capture it. Using $S = 20$ (approximate vessel segment length of 6 mm for steps of 0.3 mm) proved to be a good compromise between premature stops and false positive detections (related to jumping into veins or ventricles, mainly).

7.5. Mean-shift for result extraction

The successive generations of particles provide us with a Monte-Carlo approximation of the posterior distribution, from which we wish to extract a more compact, practical segmentation result. More precisely, we are interested in extracting the centerline curve and associated local radius. To that aim, a first possibility is to select, for each generation, the particle of highest weight [16,17]. An alternative solution, corresponding to the true use of particle filters as an estimation tool [14], is to extract the Monte-Carlo estimates of the expected centerline position $\hat{\mathbb{E}}[p_t|z_{1:t}]$ and radius value $\hat{\mathbb{E}}[r_t|z_{1:t}]$, as proposed in [1]. The second approach is arguably more sound from a theoretical point of view, leading to more stable and smoother results. It is however sensitive to asymmetric tails in the distribution, which may yield some inaccuracies by shifting expectations away from local maxima.[3] In our implementation, we adopt an intermediate solution. We simply reuse the successive mean-shift modes as an estimation of the centerline curve[4] (see Fig. 16). From a theoretical point of view, these modes correspond to the local maxima of the estimated density. Kernel regularization ensures smoother results than simply selecting the best particle, while the estimation is more local and generally slightly more accurate than the corresponding Monte-Carlo estimate. Sample results are illustrated in Section 8.

The radius estimation associated with a mean-shift mode m_t is given by:

$$r_{m_t} = \sum_{i=1}^{N_t} \hat{w}_t^i K_E\left(\frac{m_t - p_t^i}{h_t}\right) r_t^i \tag{28}$$

with K_E the Epanechnikov kernel, with the same bandwidth h_t as the one used for the mean-shift procedure. This estimate is again more local and, in our tests, slightly more accurate than the Monte-Carlo one.

Finally, one should note that our resulting representation (a chain $\{m_t, r_{m_t}\}$ of successive positions and radiuses) is not a realization of our model[5], but the result of numerical integrations over the realizations selected by the filter.

8. Experiments and evaluation

The parameter values used in our experiments are summarized in Table 2. In addition, we recall that the parameters for the histogram discretization of likelihood and transition priors are given in Table 1. Most of these parameters, from histogram discretization to approximate mean-shift, relate to approximations of “exact” results (*e.g.* the continuous distributions and the exact, full mean-shift procedure). We chose empirically the best approximations as trade-offs between memory consumption, computational performance and tracking performance. For instance, spatial parameters (h_{r_t},s) are typically chosen of the order of the CTA data resolution (0.3 mm). Histogram parameters (Table 1) are set to obtain very fine discretizations, so as not to introduce accuracy bottlenecks. We tuned these parameters on training datasets (left out of the evaluation, see Section 8.2) and kept them fixed for all the experiments, without the need for any fine-tuning for a specific dataset. We believe that the reasons for this robustness are that (i) the model combines several types of information (geometry, appearance, etc.) whose influence balance each other, (ii) the learning set was representative enough of the large variety of situations encountered in clinical routine, and (iii) some steps are specifically designed to improve robustness, such as the adaptive number of particles, or the mean-shift procedure used to extract the final decision.

Table 2
Parameter values used in the experiments.

Name	Description	Value
h_{r_t}, $h_{r_{t-1}}$	Bandwidth for kernel estimation, radius variations	0.3 mm
h_{y_t}	Bandwidth for kernel estimation, observation values	10 H.U.
h_{α_t}	Bandwidth for kernel estimation, angular variations	0.05 rad
N_p	Sub-population sampled for approximate mean-shift	5% of N_t
F	Stopping criterion (% of low likelihood particles)	25% of N_t
S	Stopping criterion (window width)	last 20 iter.
N_0	Initial number of particles	N^*
s	Discretization step to advance particles	0.3 mm

A grid search strategy could be employed to further refine these values. The most important parameter, the effective particle number N^*, is studied thoroughly hereafter.

8.1. Detailed analysis on a single dataset

To evaluate the behavior of the proposed AAPF algorithm, we first conducted a series of experiments on a single dataset with good image quality, but with a particularly complex left coronary tree, with numerous branches of various sizes and curvatures (see Fig. 19).

The evaluation criteria we use are directly inspired by those proposed in [53]. They cover both robustness (overlap with ground truth delineation) and accuracy aspects:

- FN: the average percentage of false negatives, *i.e.* the portion of the ground truth centerline delineation not successfully extracted by the algorithm;
- FP: the average percentage of false positives, *i.e.* the portion of the extracted centerline not present in the ground truth delineation;
- OV: the average centerline overlap ratio (see below);
- OT: the average centerline overlap ratio for clinically relevant vessels (diameter ≥ 1.5 mm);
- AI: the average centerline error (average distance to the ground truth, for true positive portions of the result, along with standard deviation in parentheses);
- AR: the average radius estimation error (for true positive portions of the result, along with standard deviation in parentheses);

The first four measures evaluate the robustness of the method, the last two its centerline and radius accuracy. The overlap ratio OV is defined as:

$$\mathrm{OV} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FN} + \mathrm{FP}}$$

where TP is the percentage of true positives. The ratio OV corresponds to a Dice coefficient measuring the similarity between the result and the ground truth. The ratio OT is defined similarly to OV, but restricted to clinically relevant arteries, of a diameter greater than 1.5 mm [53].

8.1.1. Target effective sample size N^* and number of allocated particles

A key parameter of our AAPF algorithm is the target effective sample size N^* used for the dynamic adaptation of the number N_t of particles allocated at each iteration (see Section 6.2). In our

[3] Since each branch is tracked independently, we expect the position and radius posterior distributions to be generally mono-modal, but not necessarily symmetric.

[4] Even if no bifurcations are detected, we extract the position of the single mode.

[5] Monte-Carlo estimates also would not guarantee that the spacing of the extracted points and their orientations satisfy exactly the properties of our geometric model.

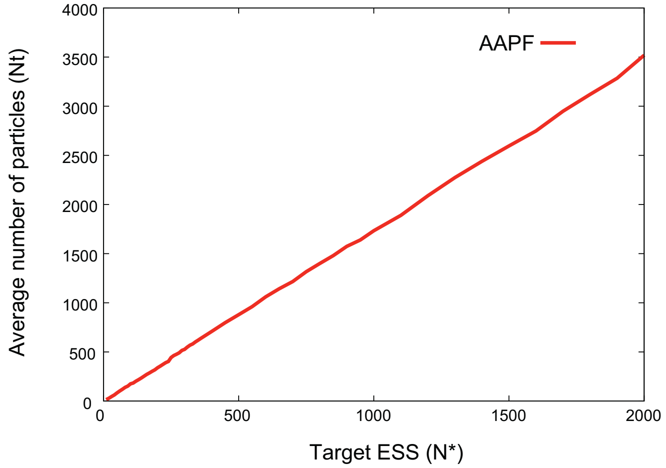

Fig. 17. Average number of particles $\bar{N}_t$ as a function of the target ESS N^*, for a CTA coronary artery tracking experiment. Illustrative result for one test dataset, for which we observed $\bar{N}_t \simeq 1.75N^*$.

tests, we observed variations of N_t between approximately $1.5N^*$ up to $5N^*$ at distal ends of thin, noisy coronary arteries. Fig. 17 depicts the average number of particles $\bar{N}_t$ as a function of N^* for the selected dataset. The relation is almost linear, with $\bar{N}_t \simeq 1.75N^*$. Tests on different datasets led to similar results, with slightly different linear slope coefficients, depending mainly on the image quality. Expectedly, lower image qualities tend to induce larger slope coefficients.

8.1.2. Comparison of SIR, APF and AAPF

The performance of our AAPF algorithm was compared with classical sampling-importance-resampling (SIR) and auxiliary particle filter (APF) implementations. All three algorithms employed explicit detection of bifurcations by mean-shift clustering (Section 7.3). For the sake of fairness, the different algorithms were compared for equivalent levels of computational cost. AAPF propagates a varying number of particles N_t at each iteration, determined with respect to the target effective sample size N^*. One iteration of the AAPF, consisting of one simulation pass and one actual propagation, is equivalent to the evaluation of $2\bar{N}_t$ particles, with $\bar{N}_t$ the average number of particles induced by N^*. The number of particles for each algorithm was thus set as follows:

- AAPF uses a target effective sample size N^*;
- APF uses $\bar{N}_t$ particles for simulation (see Section 6.1) and another $N_{APF} = \bar{N}_t$ for the actual propagation;
- SIR uses $N_{SIR} = 2\bar{N}_t$ particles.

The number of particles propagated by the APF and AAPF (on average) is thus half that of SIR.

Fig. 18 shows the average Dice coefficient of overlap with the ground truth centerline (OV) as a function of the number of particles for the three algorithms. As expected, increasing the number of particles improves the performance of all three algorithms. Interestingly, up to $N_{SIR} \simeq 2000$ particles, the performance of the SIR algorithm is superior to that of the computationally equivalent APF. Auxiliary sampling indeed improves the allocation of the samples, but for relatively low numbers of particles, it is actually advisable to sample twice as many SIR particles for the same computational cost. For $N_{SIR} > 2000$, the APF slightly outperforms classical SIR. The benefit from the dynamic adaptation of the sample size is obvious. The AAPF algorithm clearly outperforms SIR and APF, even for relatively low numbers of particles. We attribute this result to a better capture of secondary branches by the AAPF, even for relatively low values of N^* (see Section 7.3). The same result was achieved during the quantitative evaluation on the entire database (see Section 8.2). All three algorithms reach a performance plateau ($OV \simeq 0.93$ for this test dataset). As illustrated in Fig. 19 (middle), typical runs consistently fail at extracting one thin, secondary branch independently of the number of particles employed. We believe such issues to be linked to our model, as discussed in Section 8.2.

An important practical issue with stochastic approaches is the consistency of the extracted result over runs. As illustrated in Fig. 19, different runs may yield different results. Limited variations of the estimated centerline are acceptable. On the other hand, the fact that the algorithm may sometimes miss a branch entirely is much more problematic. In Fig. 20, the estimated standard deviation of the overlap OV is plotted as a function of the number of particles. For all three algorithms, a peak of the resulting variance is obtained for low numbers of particles. These peaks correspond to the points where the filters start having sufficient statistical power, in most cases, to extract significant portions of the coronary tree (see Fig. 18). We emphasize that a standard deviation of 0.16, for a Dice coefficient, is considerable. This corresponds to the loss or gain of several secondary branches. As expected, the variance decreases for larger numbers of particles. The SIR algorithm appears to have the slowest decrease rate, but APF and AAPF have slightly higher initial variances. For reasonable values of N^*, the AAPF exhibits a lower variance than both other algorithms.

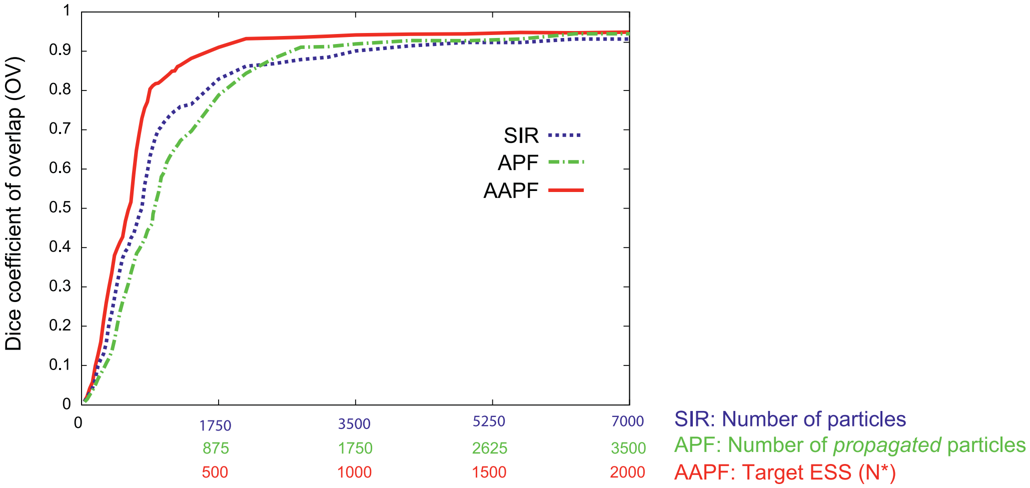

Fig. 18. Average Dice coefficient of centerline overlap with ground truth (OV) for SIR, APF and AAPF algorithms. Results averaged on 50 independent runs on a single cardiac CTA data set. Each algorithm uses a number of particles corresponding to an equivalent level of computational cost (see text).

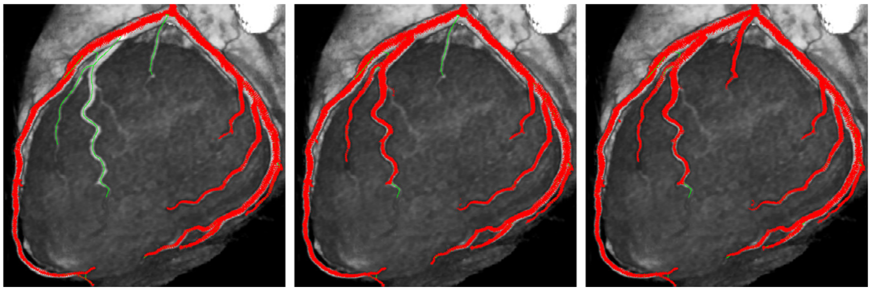

Fig. 19. Different runs of the AAPF on the same dataset ($N^* = 1000$). Left: an isolated lower quality result (OV ≤ 0.83) with several missed branches, obtained in $\sim 4\%$ of the runs. Middle: a typical run (OV $= 0.93$). Right: a high quality run (OV ≥ 0.97), obtained in $\sim 30\%$ of the runs. Ground truth centerline delineation is depicted in green. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

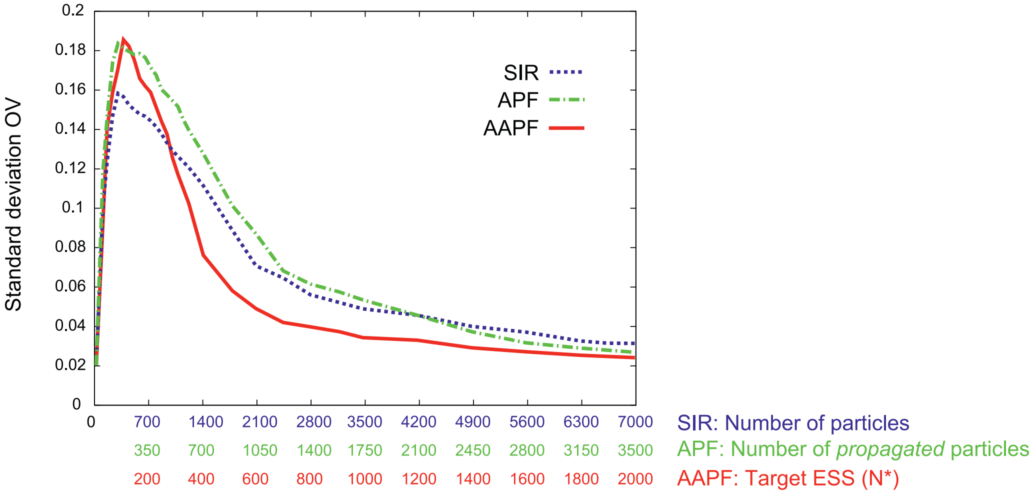

Fig. 20. Standard deviation of the Dice coefficient of overlap with the ground truth centerline (OV) for SIR, APF and AAPF algorithms. Results obtained for 50 independent runs on a single cardiac CTA data set. Each algorithm uses a number of particles corresponding to an equivalent level of computational cost (see text).

In practice, for $N^* = 1000$, the results obtained by the AAPF are fairly stable, with very little difference between runs. The variance is however non-null and the algorithm could still miss, or gain (see Fig. 19, right) a branch between runs. These variations concern mainly thin secondary branches, which account for relatively small fractions of the overall coronary trees. The main coronary branches are most generally extracted in a very consistent manner, even for relatively low values of N^*.[6]

8.1.3. *Algorithmic complexity and execution time*

In the previous sections, we showed that for an equivalent number of particle evaluations, our AAPF algorithm outperforms classical particle filters (SIR and APF). In practice, our AAPF approach can thus offer similar or better performance for a lower computational cost.

The AAPF algorithm involves only $O(N_t)$ operations and we showed in Section 8.1.1 that the observed relation between the average number of particles $\bar{N}_t$ and N^* is empirically linear, with a multiplicative constant depending on the data being segmented. Our implementation of mean-shift clustering is however of quadratic complexity $O((N_t)^2)$. Fortunately, the algorithmic refinements discussed in Section 7.3 alleviate its impact on the overall execution time. As illustrated in Fig. 21, the average execution time increases in a non-linear, but less than quadratic, function of N^*. Note that for low values of N^* ($N^* < 400$), the execution time is understandably very low as the filter is generally unable to explore the entire vascular tree.

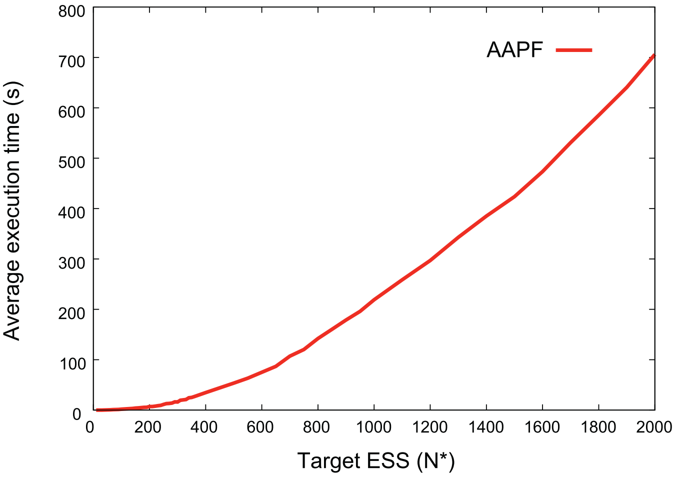

Fig. 21. Average execution time as a function of N^*. Results were averaged on 50 independent runs for a single dataset.

Despite our refinements, the computational overhead induced by the use of mean-shift clustering remains relatively high. Bifurcation detection accounts for approximately 20% of the overall execution time. We emphasize however that our approach remains reasonably fast, thanks to the computational efficiency of the underlying components (`MFlux` feature in particular). For $N^* = 1000$, a full coronary tree is generally extracted within 4 minutes (2.16 GHz `Intel Core Duo Intel` processor). Given the numerous possibilities for

[6] In our tests, a value of $N^* = 500$ seemed sufficient for the reliable extraction of the main coronary arteries.

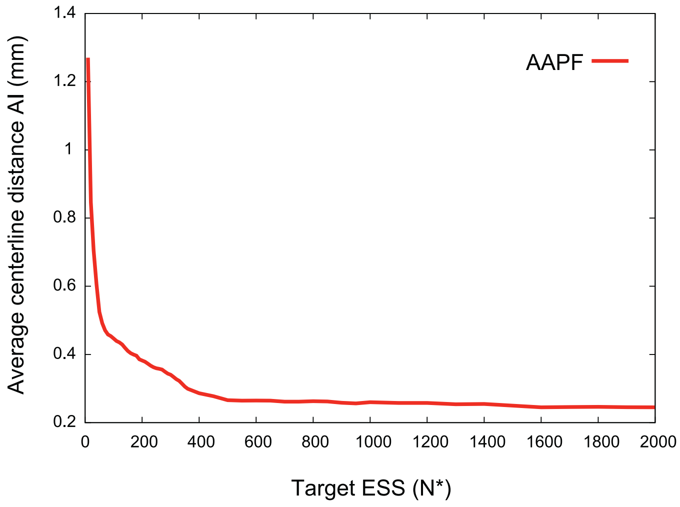

Fig. 22. Average centerline distance (AI) to the ground truth as a function of N^*. Result were averaged on 50 independent runs for a single dataset.

improvement (massive parallelization and optimization of mean-shift clustering), we believe that the performance of our proof-of-concept implementation is promising.

8.1.4. Centerline accuracy

As depicted in Fig. 22, the average centerline error (AI) decreases relatively quickly with increasing N^*. For $N^* \simeq 600$, AI reaches $\simeq 0.26\,\text{mm}$, which is a satisfyingly subvoxelic precision (the intra-slice resolution of the test dataset is 0.33 mm). Increasing N^* further brings only marginal improvement of the centerline accuracy (0.248 mm for $N^* = 2000$).

8.2. Evaluation on a CTA database

Our algorithm was quantitatively evaluated on our ground truth database of manually delineated cardiac CTA datasets. We used 10 randomly selected datasets for training (*i.e.* learning the likelihood and prior distributions) and the 51 remaining ones for testing. This database presents a lot of variability, representative of the variety of situations that can be encountered in clinical routine. The ground-truth data consist of comprehensive segmentations of the entire left and right coronary trees (not just a few main branches). These segmentations were performed manually by experts and validated by a radiologist. So we can evaluate the extraction of both left and right entire coronary trees, which was not possible with existing databases. Sample results are given in Figs. 23 and 24.

Table 3 summarizes the quantitative results for the SIR, APF and AAPF algorithms. These results were obtained for $N^* = 1000$ for the AAPF, ensuring sufficient statistical power and low variance of the results. For each dataset, we first ran the AAPF, computed the average number $\bar{N}_t$ of particles used, and ran the SIR algorithm with $N_{SIR} = 2\bar{N}_t$ particles and the APF with $N_{APF} = \bar{N}_t$.

Table 3
Quantitative validation on 51 datasets. Results averaged on 20 runs for each dataset. See text for details.

Measure	SIR $(N_{SIR} = 2\bar{N}_t)$	APF $(N_{APF} = \bar{N}_t)$	AAPF $(N^* = 1000)$
FN	21.6%	19.3%	15.4%
FP	12.5%	12.1%	12.3%
OV	83%	84.3%	86.2%
Std. dev.OV	4.7%	3.9%	3.2%
OT	92.1%	92.4%	92.5%
Std. dev.OT	1.7%	1.7%	1.5%
AI (mm)	0.26 (±0.22)	0.26 (±0.21)	0.25 (±0.21)
AR (mm)	0.21 (±0.19)	0.20 (±0.18)	0.20 (±0.18)

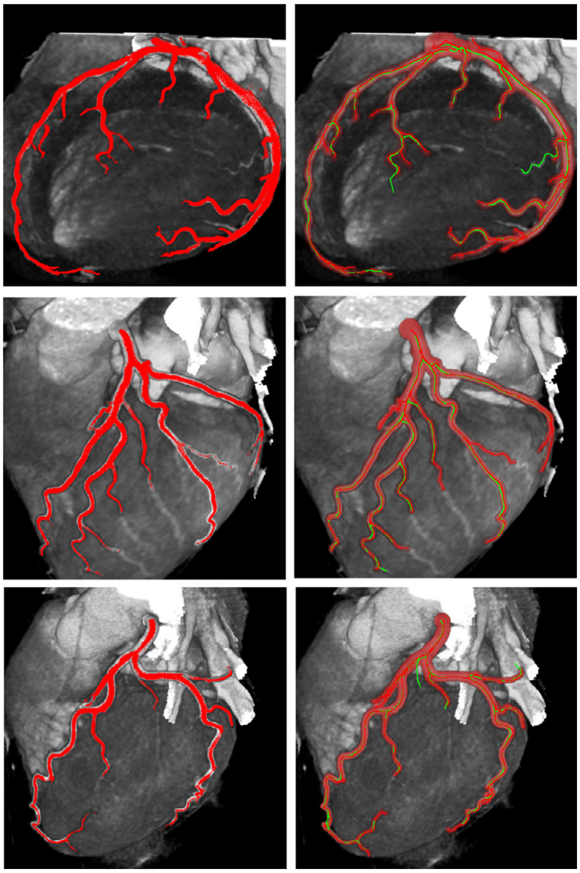

Fig. 23. Tracking with particle filtering: result samples for left coronary trees. Left: successive generations of particles. Right: the extracted centerline is given in red. The associated mask (light red) is obtained from the corresponding radius estimation. The ground truth centerline is depicted as a green curve. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

8.2.1. Comparison of AAPF, APF and SIR

The quantitative results confirm the experiments, with AAPF outperforming APF and SIR in terms of robustness (overlap criteria) and leading to a marginal improvement of centerline accuracy. The AAPF approach notably yields an appreciable decrease in the rate of false negatives (missed segments) and inter-run variance, confirming that our adaptive method is able to explore the arterial tree in a more consistent and exhaustive manner.

8.2.2. Overlap measures (OV and OT)

In terms of overlap, our AAPF approach demonstrates its high robustness with an average OV percentage of 86.2%. Errors are dominated by the rate of false negatives (15.4%), explained by the non-retrieval of some secondary asymmetric bifurcations. Missing or recovering such branches also explains the average inter-run standard deviation of *OV* (3.2%). This issue is discussed in more depth in a subsequent paragraph. We highlight, however, the good consistency of the result for the main coronary branches (diameter > 1.5 mm), with an overlap Dice coefficient *OT* of 92.5% and with low inter-run deviation (1.5%)

8.2.3. Centerline and radius accuracy

From a qualitative point of view, results extracted thanks to the modes of the mean-shift procedure are satisfyingly regular, following

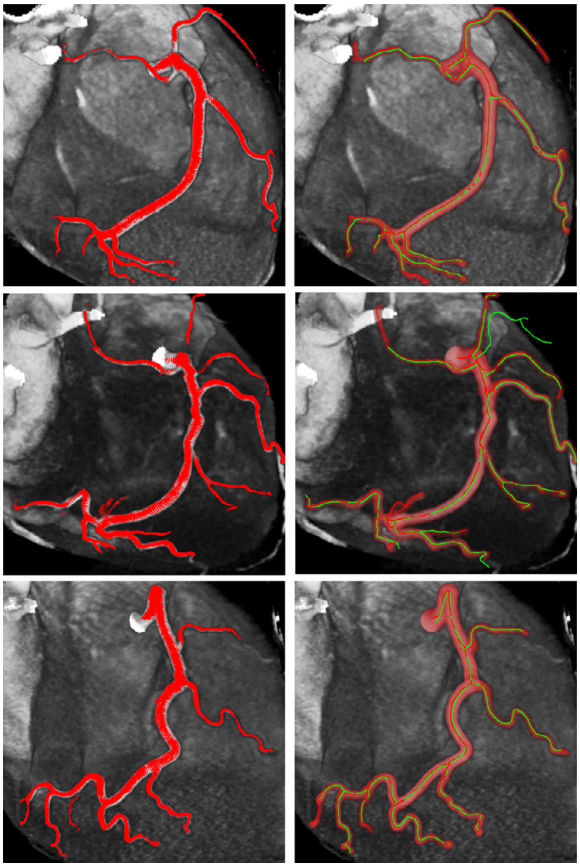

Fig. 24. Tracking with particle filtering: result samples for left coronary trees. Left: successive generations of particles. Right: the extracted centerline is given in red. The associated mask (light red) is obtained from the corresponding radius estimation. The ground truth centerline is depicted as a green curve. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

closely the ground truth delineations even in curvy portions of the coronary arteries. The algorithm can also be seen as adapting well to radius variations (see the right columns of Figs. 23 and 24). Centerline and radius errors are satisfyingly sub-voxelic (average data resolution of 0.3 mm^3), confirming the previous results. Increasing the parameter N^* over 1000 only resulted in marginal improvements in our tests, suggesting that this level of accuracy is probably more dependent on the low-level MFlux feature, and notably on the assumption of circular cross-sections, than on the extraction scheme.

8.2.4. On asymmetric bifurcations

One of the main issues observed with our AAPF algorithm is that, regardless of the number of particles employed, the filter often misses some of the smallest coronary branches, arising from asymmetric bifurcations. These secondary arteries generally branch from much bigger vessels, at relatively high angles. Such occurrences are in fact poorly predicted by our direction and scale priors, which correspond to relatively smooth, limited angular and scale variations along the vessel. One can note, in the left columns of Figs. 23 and 24, that some of the secondary branches explored by the filter are originally discovered by a small number of particles, predicted in the tails of our transition priors. For most of these branches, adequate statistical power, *i.e.* a sufficiently high number of sampled particles, is enough to guarantee their consistent detection. This is not the case, however, for those with particularly high scale variations (*e.g.* a very small coronary branching off a big one) *and* high angles of bifurcation. We believe that this is not a problem of sample impoverishment, but a more fundamental issue of our model. The algorithm has difficulties capturing some branches because it rarely samples them at all. Because the performance of the prior is in question, Markov Chain Monte Carlo (MCMC) rejuvenation, as proposed in [1] for the re-population of bifurcations, would not solve the theoretical issue at hand. MCMC steps cope with particle impoverishment, but still rely on the same transition prior as the main filtering process. Practical improvement in our case could empirically be obtained by artificially relaxing the transition prior, allowing broader angular and scale variations, but at the risk of making the exploration of the search space less efficient and more prone to false positives. Alternatively, one could probably devise a reseeding procedure to recover missed branches by exploring the immediate surroundings of the initial result. From a theoretical point of view, an interesting lead would be the use of *model-switching* approaches [33] that would allow the algorithm to take into account an explicit model for asymmetric bifurcations. The general idea would be to design a model-switching process enabling at each iteration the prediction of a certain quantity of *branching* vessels following a different transition prior (lower scales and high branching angle). Adequate switching and second transition priors could potentially be learned as in our current model.

9. Discussion and perspectives

In this paper, we have presented an optimized particle filtering approach for the extraction of full coronary trees from cardiac CTA data. This method relies on the stochastic propagation of a population of spatial point samples along the coronary arteries following a centerline-based tracking design. A key difference with classical deterministic tracking schemes is that particle filters track the entire posterior distribution of the vessels, considering numerous realizations in parallel from which an approximation of the distributions and estimates of interest are computed.

Our approach relies on a Bayesian model combining data likelihood with radius and direction priors. Our entire Bayesian model is learned non-parametrically from a ground-truth database of manually delineated datasets, using kernel density estimations, thus avoiding the delicate use of parametric distributions.

With respect to particle filter-based estimation, we introduced an adaptive sampling scheme, which we referred to as Adaptive Auxiliary Particle Filtering (AAPF). AAPF combines classical auxiliary filtering (APF) with a dynamic adaptation of the number of samples. In our application, we showed that, compared to classical particle filters, this algorithm improves the behavior at bifurcations and lowers the stochastic inter-run variance of the results. We also discussed the use of mean-shift clustering for the explicit detection of bifurcations and for the extraction of the final results. Our quantitative validation demonstrated the robustness of this approach. The proposed method can be seen, thanks to its robustness as complementary with other existing techniques. For instance it could be followed by completion with a minimal path technique, such as in [43,45], when one specifically needs to extend or correct a particular branch.

We believe that the results obtained by our proof-of-concept implementation are particularly promising. From a theoretical point of view, we mentioned the perspective of using a model-switching procedure [33] to improve the exploration of small, secondary branches. Another theoretical lead for improvement would be the use of smoothing estimation schemes [14,32,36], *i.e.* estimations taking into account future observations (in opposition to filtering techniques, which rely solely on past observations). The idea of using future observations is particularly attractive in our applicative context, where the observation source (the image) is static and fully accessible. Smoothing techniques are considerably more computationally demanding than their filtering counterparts, but recent developments have shown promising optimization techniques [37]. An alternative, which could integrate directly within our framework, would be the use of dynamic programming methods to extract the MAP sequence from successive particle generations [26]. Such a refinement would also induce considerable computational overhead.

Finally, we emphasize that our current implementation could be optimized in numerous ways. One could first accelerate mean-shift clustering thanks to some of the various existing refinements [9,19,23,61,65]. A second, highly promising lead for optimization is the parallelization of the algorithm. In fact, most operations involved in our approach can be straightforwardly parallelized. In our tests, we obtained a 35% performance gain by simply parallelizing the execution of the filter on two processor cores (`OpenMP` directives on a `Intel Core Duo Intel` processor). We believe that a massively parallel GPU implementation could reach very high levels of computational efficiency.

References

[1] K. Allen, C. Yau, J. Noble, A recursive, stochastic vessel segmentation framework that robustly handles bifurcations, in: Proceedings of Medical Image Understanding and Analysis, 2008.

[2] M. Arulampalam, S. Maskell, N. Gordon, T. Clapp, D. Sci, T. Organ, S. Adelaide, A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking, IEEE Trans. Signal Process. 50 (2) (2002) 174–188.

[3] N. Bergman, Recursive Bayesian Estimation: Navigation and Tracking Applications Dissertation no. 579. Ph.D. thesis, Linköping Studies in Science and Technology, 1999.SE-581

[4] H. Blum, A transformation for extracting new descriptors of shape, in: Models for the Perception of Speech and Visual Form, 1967, pp. 362–380.

[5] H. Blum, Biological shape and visual science. I, J. Theor. Biol. 38 (2) (1973) 205–287.

[6] M. Brady, H. Asada, Smoothed local symmetries and their implementation, Int. J. Robot. Res. 3 (3) (1984) 36.

[7] G. Casella, C. Robert, Rao-Blackwellisation of sampling schemes, Biometrika 83 (1) (1996) 81–94.

[8] D. Comaniciu, An algorithm for data-driven bandwidth selection, IEEE Trans. Pattern Anal. Mach. Intell. 25 (2) (2003) 281–288.

[9] D. Comaniciu, P. Meer, Mean shift: a robust approach toward feature space analysis, IEEE Trans. Pattern Anal. Mach. Intell. 24 (5) (2002) 603–619.

[10] D. Comaniciu, V. Ramesh, P. Meer, The variable bandwidth mean shift and data-driven scale selection, Proceedings of IEEE International Conference on Computer Vision vol. 1, (2001) 438–445.

[11] T. Deschamps, L. Cohen, Minimal paths in 3D images and application to virtual endoscopy, in: Proceedings of European Conference on Computer Vision, 2000, pp. 543–560.

[12] M. Descoteaux, D.L. Collins, K. Siddiqi, A multi-scale geometric flow for segmenting vasculature in MRI, in: Proceedings of International Conference on Computer Vision and Mathematical Methods in Medical and Biomedical Image Analysis, ECCV, 2004, pp. 169–180.

[13] E. Dikici, T. O'Donnell, L. Grady, R. Setser, R. White, Coronary artery centerline tracking using axial symmetries, Insight J. (2008). http://hdl.handle.net/1926/1338.

[14] A. Doucet, N. de Freitas, N. Gordon, Sequential Monte Carlo Methods in Practice, Springer-Verlag, 2001.

[15] A. Doucet, S. Godsill, C. Andrieu, On sequential Monte Carlo sampling methods for Bayesian filtering, Stat. Comput. 10 (3) (2000) 197–208.

[16] C. Florin, N. Paragios, J. Williams, Particle filters, a quasi-Monte Carlo solution for segmentation of coronaries, in: Proceedings of Medical Image Computing and Computer Assisted Intervention, 2005, pp. 246–253.

[17] C. Florin, N. Paragios, J. Williams, Globally optimal active contours, sequential Monte-Carlo and on-line learning for vessel segmentation, in: Proceedings of European Conference on Computer Vision, 2006, pp. 476–489.

[18] D. Fox, Adapting the sample size in particle filters through KLD-sampling, Int. J. Robot. Res. 22 (12) (2003) 985.

[19] D. Freedman, P. Kisilev, Fast mean shift by compact density representation, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2009.

[20] O. Friman, M. Hindennach, C. Kuehnel, H.O. Peitgen, Non-stochastic sampling for multiple hypotheses, Med. Image Anal. 14 (2) (2010) 160–171.

[21] K. Fukunaga, L. Hostetler, The estimation of the gradient of a density function, with applications in pattern recognition, IEEE Trans. Inf. Theory 21 (1) (1975) 32–40.

[22] D. Geman, B. Jedynak, An active testing model for tracking roads in satellite images, IEEE Trans. Pattern Anal. Mach. Intell. 18 (1) (1996) 1–14.

[23] B. Georgescu, I. Shimshoni, P. Meer, Mean shift based clustering in high dimensions: a texture classification example, in: Proceedings of IEEE International Conference on Computer Vision, 2003, pp. 456–463.

[24] J. Geweke, Bayesian inference in econometric models using Monte Carlo integration, Econom.: J. Econom. Soc. 57 (6) (1989) 1317–1339.

[25] S. Godsill, T. Clapp, Improvement strategies for Monte Carlo particle filters, Sequential Monte Carlo Methods in Practice, Springer-Verlag, 2001, pp. 139–158.

[26] S. Godsill, A. Doucet, M. West, Maximum a posteriori sequence estimation using Monte Carlo particle filters, Ann. Inst. Stat. Math. 53 (1) (2001) 82–96.

[27] N. Gordon, D. Salmond, A. Smith, Novel approach to nonlinear/non-Gaussian Bayesian state estimation, Proc. Inst. Elect. Eng. 140 (1993) 107–113.

[28] L. Grady, Fast, quality, segmentation of large volumes-isoperimetric distance trees, in: Proceedings of European Conference on Computer Vision, ECCV, vol. 3953, 2006, pp. 449–462.

[29] L. Grady, E. Schwartz, Isoperimetric graph partitioning for image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 28 (3) (2006) 469–475.

[30] M.A. Gulsun, H. Tek, Robust vessel tree modeling, in: Proceedings of Medical Image Computing and Computer-Assisted Intervention, 2008.

[31] M. Holtzman-Gazit, R. Kimmel, N. Peled, D. Goldsher, Segmentation of thin structures in volumetric medical images, IEEE Trans. Image Process. 15 (2) (2006) 354–363.

[32] M. Hürzeler, H. Künsch, Monte Carlo approximations for general state-space models, J. Comput. Graph. Stat. 7 (2) (1998) 175–193.

[33] M. Isard, A. Blake, A mixed-state condensation tracker with automatic model-switching, in: Proceedings of IEEE International Conference on Computer Vision, 1998, pp. 107–112.

[34] M. Isard, A. Blake, Condensation: conditional density propagation for visual tracking, Int. J. Comput. Vis. 29 (1998) 5–28.

[35] R. Kimmel, A.M. Bruckstein, Regularized laplacian zero crossings as optimal edge integrators, Int. J. Comput. Vis. 53 (3) (2003) 225–243.

[36] G. Kitagawa, Monte Carlo filter and smoother for non-Gaussian nonlinear state space models, J. Comput. Graph. Stat. 5 (1) (1996) 1–25.

[37] M. Klaas, M. Briers, N. de Freitas, A. Doucet, S. Maskell, D. Lang, Fast particle smoothing: if i had a million particles, in: Proceedings of IEEE International Conference on Machine Learning, 2006, pp. 481–488.

[38] T. Koller, G. Gerig, G. Székely, D. Dettwiler, Multiscale detection of curvilinear structures in 2D and 3D image data, in: Proceedings of IEEE International Conference on Computer Vision, 1995, pp. 864–869.

[39] S. Konishi, A. Yuille, J. Coughlan, S. Zhu, Statistical edge detection: learning and evaluating edge cues, IEEE Trans. Pattern Anal. Mach. Intell. 25 (1) (2003) 57–74.

[40] O. Lanz, An information theoretic rule for sample size adaptation in particle filtering, in: Proceedings of International Conference on Image Analysis and Processes, 2007, pp. 317–322.

[41] D. Lesage, A. Angelini, I. Bloch, G. Funka-Lea, A review of 3d vessel lumen segmentation techniques: models, features and extraction schemes, Med. Image Anal. 13 (2009) 819–845.

[42] D. Lesage, E. Angelini, I. Bloch, G. Funka-Lea, Medial-based Bayesian tracking for vascular segmentation: application to coronary arteries in 3d CT angiography, in: Proceedings of IEEE International Symposium on Biomedical Imaging, 2008, pp. 268–271.

[43] D. Lesage, E. Angelini, I. Bloch, G. Funka-Lea, Bayesian maximal paths for coronary artery segmentation from 3d CT angiograms, in: Proceedings of Medical Image Computing and Computer Assisted Intervention, 2009, pp. 222–229.

[44] D. Lesage, E. Angelini, I. Bloch, G. Funka-Lea, Design and study of flux-based features for 3d vascular tracking, in: Proceedings of IEEE International Symposium in Biomedical Imaging, 2009, pp. 286–289.

[45] D. Lesage, Models, Features and Extraction Schemes for Vascular Segmentation: Application to the delineation of Coronary Arteries from 3D Computed Tomography Data Ph.D. thesis, Telecom ParisTech, 2009.No. 2009E044

[46] H. Li, A. Yezzi, Vessels as 4-D curves: global minimal 4-D paths to extract 3-D tubular surfaces and centerlines, IEEE Trans. Med. Imag. 26 (2007) 1213–1223.

[47] J. Liu, R. Chen, Sequential Monte Carlo methods for dynamical systems, J. Am. Stat. Assoc. 93 (1998) 1032–1044.

[48] R. Van der Merwe, A. Doucet, N. De Freitas, E. Wan, The unscented particle filter, Adv. Neural Inform. Process. Syst. (2001) 584–590.

[49] B. Nayebifar, H.A. Moghaddam, A novel method for retinal vessel tracking using particle filters, Comput. Biol. Med. 43 (5) (2013) 541–548.

[50] N. Oudjane, C. Musso, Progressive correction for regularized particle filters, in: Proceedings of International Conference on Information Fusion, vol. 2, 2000.

[51] M. Pitt, N. Shephard, Filtering via simulation: auxiliary particle filters., J. Am. Stat. Assoc. 94 (446) (1999) 590–591.

[52] M. Schaap, R. Manniesing, I. Smal, T. van Walsum, A. van der Lugt, W.J. Niessen, Bayesian tracking of tubular structures and its application to carotid arteries in cta, in: Proceedings of Medical Image Computing and Computer-Assisted Intervention, 2007, pp. 562–570.

[53] M. Schaap, C. Metz, T. van Walsum, A. van der Giessen, A. Weustink, N. Mollet, C. Bauer, H. Bogunovic', C. Castro, X. Deng, E. Dikici, T. O'Donnell, M. Frenay, O. Friman, M.H. Hoyos, P. Kitslaar, K. Krissian, C. Khnel, M.A. Luengo-Oroz, M. Orkisz, Smedby, M. Styner, A. Szymczak, H. Tek, C. Wang, S.K. Warfield, S. Zambal, Y. Zhang, G.P. Krestin, W. Niessen, Standardized evaluation methodology and reference database for evaluating coronary artery centerline extraction algorithms, Med. Image Anal. 13 (5) (2009) 701–714.

[54] M. Schaap, I. Smal, C. Metz, T. van Walsum, W. Niessen, Bayesian tracking of elongated structures in 3D images, Proceedings of International Conference on Information Processing in Medical Imaging, vol. 4584, Springer, 2007, pp. 74–85.

[55] H. Shim, D. Kwon, I.D. Yun, S.U. Lee, Robust segmentation of cerebral arterial segments by a sequential Monte Carlo method: particle filtering, Comput. Methods Programs Biomed. 84 (2–3) (2006) 135–145.

[56] B. Silverman, Density Estimation for Statistics and Data Analysis, Chapman & Hall, 1986.

[57] A. Soto, Self adaptive particle filter, in: Proceedings of International Conference on Artificial Intelligence, vol. 19, 2005, p. 1398.

[58] O. Straka, M. Simandl, Particle filter adaptation based on efficient sample size, in: Proceedings of IFAC Symposium on System Identification, 2006, pp. 991–996.

[59] H. Tek, M. Gulsun, S. Laguitton, L. Grady, D. Lesage, G. Funka-Lea, Automatic coronary tree modeling, Insight J. (2008).

[60] A. Vasilevskiy, K. Siddiqi, Flux maximizing geometric flows, IEEE Trans. Pattern Anal. Mach. Intell. 24 (12) (2002) 1565–1578.

[61] A. Vedaldi, S. Soatto, Quick shift and kernel methods for mode seeking, Proceedings of European Conference on Computer Vision, Springer, 2008.

[62] S. Wang, B. Peplinski, L. Lu, W. Zhang, J. Liu, Z. Wei, R.M. Summers, Sequential Monte Carlo tracking for marginal artery segmentation on CT angiography by multiple cue fusion, in: Medical Image Computing and Computer Assisted Intervention, in: LNCS, vol. 8150, 2013, pp. 518–525.

[63] X. Wang, T. Heimann, P. Lo, M. Sumkauskaite, M. Puderbach, M. de Bruijne, H.P. Meinzer, I. Wegner, Statistical tracking of tree-like tubular structures with efficient brandching detection in 3D medical image data, Phys. Med. Biol. 57 (16) (2012) 5325–5342.

[64] S. Woerz, W.J. Godinez, K. Rohr, Segmentation of 3D tubular structures based on 3D intensity models and particle filter tracking, in: Proceedings of SPIE Medical Imaging, vol. 7259, 2009.

[65] C. Yang, R. Duraiswami, N. Gumerov, L. Davis, Improved fast Gauss transform and efficient kernel density estimation, in: Proceedings of IEEE International Conference on Computer Vision, 2003, pp. 664–671.

[66] A. Yuille, J. Coughlan, Twenty questions, focus of attention, and A∗: a theoretical comparison of optimization strategies, in: Proceedings of Energy Minimization Methods in Computer Vision and Pattern Recognition, 1997, p. 197.

[67] A. Yuille, J. Coughlan, Fundamental limits of Bayesian inference: order parameters and phase transitions for road tracking, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2) (2000) 160–173.

[68] F. Zhao, R. Bhotika, P. Mendonca, N. Krahnstoever, J.V. Miller, Adaptive intensity models for probabilistic tracking of 3D vasculature, in: Proceedings of IEEE International Symposium in Biomedical Imaging, 2010, pp. 41–44.