4

From image data to computational domains

Luca Antiga, Joaquim Peiró, and David A. Steinman

The advent of high-resolution imaging systems and powerful computational resources has made it possible to obtain information about the *in vivo* anatomy of blood vessels in a non invasive way. By employing this information as the domain definition for computational fluid dynamics, it is now possible to model hemodynamics in realistic geometric configurations on a subject-specific basis. Since geometry has a strong influence on hemodynamics, as will shown extensively in Chapter 5, the procedure used to model the geometry of a blood vessel from medical images plays a primary role in determining the reliability of haemodynamic predictions and, ultimately, their clinical significance.

In this chapter we will describe the process of defining domains for hemodynamics modelling from clinically acquired images. The solution of the equations of flow, transport and structure (see Chapter 2 and 3). in realistic geometries requires the use of numerical approximation methods, such as finite elements and finite volumes [371, 407]. Such methods are based on the discretisation of the physical domain in elements of simple shape and finite size (e.g. tetrahedra or hexahedra), which constitute the so-called computational mesh. The process from images to computational meshes representing realistic vascular geometries involves several separate steps (which are summarised in the flow chart of Fig. 4.1): i) Sets of images are first acquired using one of the clinically available imaging techniques. Imaging datasets typically come in the form of sets of 2D images each having a well defined position and orientation in space. Such sets can either be formed by contiguous images that can be stacked along a spatial direction, thus defining a volume, or be constituted by disjoint images. In addition, series of images of the same anatomical structure can be taken at several instants in time, effectively increasing the dimensionality of the data-set. ii) According to image quality and acquisition technique characteristics, images eventually undergo an enhancement step, during which vascular structures are made clearer at the expense of noise and non-vascular content. iii) Image segmentation, which is the process by which the regions of the belonging to vascular structures are identified, is then performed. Depend-

Formaggia L, Quarteroni A, Veneziani A. (Eds.): Cardiovascular Mathematics. Modeling and simulation of the circulatory system

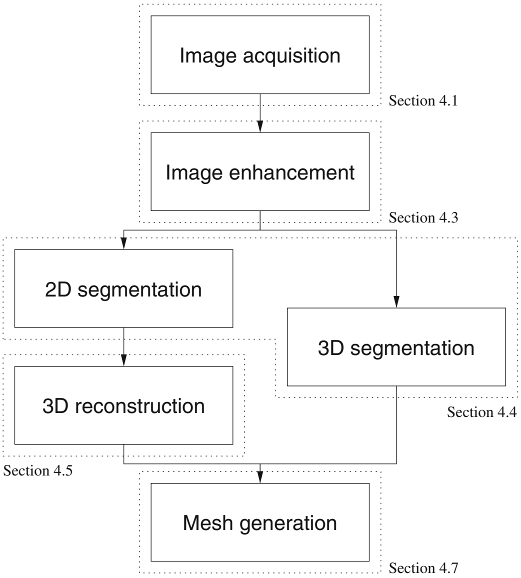

Fig. 4.1. Flow chart depicting the typical workflow from medical images to computational meshes. References to sections in the present chapter covering the enclosed topics are also reported

ing on the nature of the source images and on the geometric complexity of the 3D vascular structure of interest, segmentation can take place on single 2D planes or directly in 3D. In the first case, a 3D reconstruction phase is needed in order to generate the final 3D shape of the vessel from the information defined on the separate 2D planes. Finally, in case of time-resolved acquisitions, the segmentation have to adapt to the changes in shape occurring over time. iv) Finally, the definition of the vessel surface is employed for the generation of the computational mesh onto which the hemodynamics modelling problem can be solved numerically. For this purpose, well-shaped 2D elements are generated over the surface, and the model interior is then filled with 3D elements of appropriate density for the haemodynamic modelling problem.

While in the past several works have dealt with the description of the process from images to computational models as a whole, the tendency of current methodological publications on the subject is to focus on particular aspects of the process. Indeed, the way the single steps of the process are performed is potentially interchangeable (e.g. image segmentation is typically independent of mesh generation), and not all the possible combinations of techniques have actually been documented within a description of the whole process. We will therefore try to draw a sketch of the most relevant techniques today available for generating subject-specific models of blood vessels from medical images, including, for each step, the techniques that have been or could be potentially employed for the purpose.

4.1 Vascular imaging

Before reviewing the techniques involved in turning medical image data into geometries, it is important to appreciate the underlying physics, capabilities and limitations of the three main vascular imaging modalities available: X-ray, i.e. digital subtraction angiography and computerised tomography [206,284,389], magnetic resonance [203,327,369], and ultrasound [45,144,228] imaging. In the interest of space these are only briefly discussed below; for further information the reader is referred to the recent reviews cited. In general, however, it should be appreciated throughout that superior image quality and resolution invariably come at the expense of invasiveness and acquisition time.

4.1.1 X-ray imaging

Akin to casting a shadow, x-ray imaging records the projection of x-ray beams through the body and onto a radiographic film, fluoroscopic screen, or digital detector. Differential absorption of x-rays by the various body structures produces contrast in the resulting 2D images, though it is usually only possible to discriminate among bone, soft tissue and air. Highlighting the vascular lumen anatomy at the expense of the wall and surrounding tissue – angiography – typically requires the introduction of a radio-opaque dye into the blood stream.

Traditionally, x-ray angiography highlights the lumen by injecting an iodine-based contrast agent via a catheter placed directly into the artery of interest, which serves to effectively replace the blood within the artery for a few seconds. This results in superior contrast, and so permits the real-time acquisition of the projections. Such "fluoroscopic" imaging is particularly suited to mobile vessels like the coronary arteries (see Fig. 4.2 A), and so can be useful for prescribing time-varying motions for blood flow simulations (see Chapter 9). For relatively static vessels like the cerebral or peripheral arteries, digital subtraction angiography (DSA), which acquires an image prior to the contrast agent injection and then subtracts it digitally, can be used to better discriminate the opacified lumen from surrounding bone and soft tissue. Either way, real time or at least high temporal resolution (cine) angiography allows for the visualisation of contrast agent as it flows into the vascular tree. Although in principle this makes it possible to quantify blood flow rates and residence times [463], in practise it is most often used clinically to identify late or absent filling of distal and collateral vessels.

Owing to its superior contrast and spatial resolution ($\sim 0.2\,\text{mm}$), projection x-ray angiography remains the gold standard for most vascular imaging applications. Still, its projective nature means that often-tortuous 3D lumen geometries can only be properly reconstructed from multiple projections, typically coupled with simplifying assumptions about the shape of the lumen cross-section (see Section 4.5). Such assumptions are unsuitable for truly 3D vascular structures like cerebral aneurysms (Fig. 4.2 B), so to get around this, rotational angiography techniques (RA) have been developed in which the gantry supporting the x-ray source and detector is rotated rapidly around the patient, allowing for the acquisition of many projections within the few seconds that contrast agent is being injected. From these multiple projections it is then possible to reconstruct high quality, high resolution (~ 0.4 mm isotropic, meaning the same resolution in all three directions) 3D images of the artery.

As with all catheterisation procedures, intra-arterial angiography carries with it a small but non-negligible risk of emboli (blood clots, air bubbles, etc.) travelling downstream and getting lodged in smaller arteries, a risk often higher than that associated with ionising radiation itself [36]. A less invasive alternative is to inject the contrast agent intravenously – any emboli must first pass through the lungs, where, for air bubbles especially, they can be filtered out. The downside is that the contrast agent mixes with the rest of the blood volume before it reaches the arteries via the heart. This serves to dilute its effect, and so higher contrast agent or x-ray doses may be required. Intravenous injection also restricts the temporal window available for imaging: too early and the contrast agent has not yet reached the artery of interest; too late and adjacent veins become enhanced, which ultimately makes it difficult to discriminate the arteries from veins.

While projection angiography still relies on intra-arterial injection, computed tomography (CT) angiography takes advantage of intravenous injections. Building upon the principles of computed axial tomography (i.e., CAT scans), a ring incorporating one or more x-ray sources and opposing detectors is rotated rapidly around the patient, producing and then reconstructing into an image the projections from multiple fan beams. By moving the patient axially through the donut-shaped scanner during this process, volumetric (3D) images can be reconstructed (Fig. 4.2 C). Owing to the short temporal window available for purely arterial enhancement, such “helical” CTA has typically been forced to sacrifice axial for in-plane resolution by using relatively long spiral pitches, resulting in anisotropic spatial resolutions despite the nominal 3D nature of the image data (see Section 4.3). The recent availability of scanners with multiple rings of sources and detectors now makes it possible to achieve sub-millimetre isotropic resolutions. With appropriate gating (i.e., synchronisation) of the acquisition to the heartbeat, temporally-resolved 3D imaging – often referred to as 4D CT – becomes feasible. This makes possible the reconstruction of time-resolved 3D lumen boundaries, or the validation of motions predicted by fluid-structure interaction models (see Chapter 9).

4.1.2 Magnetic Resonance Imaging (MRI)

MRI exploits the phenomenon of nuclear magnetic resonance, whereby atomic nuclei possessing spin can be made to resonate at their so-called Larmor frequency, proportional to the strength of an applied magnetic field. In MRI, protons, which resonate at radiofrequencies (RF) for clinical-strength scanners, are first aligned by placing the patient within a powerful superconducting magnet. (For this reason, MRI is contraindicated for patients with metallic or electronic implants; getting within metres of an MRI scanner can also stop your watch and erase your bank cards.) Brief application of RF energy sets the protons resonating, after which the RF signals decay at rates characteristic to the tissue in which the protons are bound. By simultaneously applying magnetic field gradients, it becomes possible to effectively encode position into the frequency of the signal. As such, MR images are actually acquired in frequency space (often referred to as k-space), and images are obtained by inverse Fourier transforms. Since the strength and evolution of the MR signal are related to the tissue properties, contrast can be generated by judicious selection and timing of the RF pulses and magnetic field gradients, collectively referred to as a pulse sequence. Thus the chief advantage of MRI is that it can achieve soft tissue contrast far superior to other imaging modalities. On the other hand, since the MRI signal is relatively weak, usually sensitive RF coils (i.e., antennae) must be placed close to the body surface, often to the unease of patients already anxious after being placed within a long cylindrical magnet (c.f., the shorter, donut-shaped configuration used for CT). Because signal strength is proportional to the voxel volume – a voxel is the 3D version of a 2D image pixel – high in-plane resolutions are usually only achievable by acquiring 2D images over relatively thick slices, which has implications on the way in which the 3D vascular geometries can be reconstructed, as will be explained throughout the chapter (see Sections 4.3 and 4.5). Even then, relatively long scan times are often required to traverse k-space: a single image can take up to several minutes depending on the type of contrast required. Patient motion then becomes a major challenge, especially since acquisition in frequency space can produce non-intuitive artifacts in the transformed image.

For vascular imaging, MRI has an important advantage over x-ray-based techniques: contrast can be generated without the need for exogenous agents. In time-of-flight (TOF) MRA angiography, static tissue is suppressed by saturating it with RF energy within a desired slice or volume, and then unsaturated blood flows in, appearing bright against the suppressed static tissue (Fig. 4.2 D). The timing of TOF angiograms must therefore be done carefully, unsaturated blood must be allowed enough time to flow in, but not so much that it becomes suppressed itself. This can be a problem for volumetric (3D) acquisitions, and so, even though less time-efficient, TOF angiograms are often acquired as a series of contiguous or overlapping thick (2D) axial slices. Still, artefactual signal voids can appear even in 2D images when complex, recirculating blood flow is present, confounding the segmentation of the lumen boundaries. Care must also be exercised in choosing the image resolution: too coarse and the blood signal is weakened by so-called phase dispersion caused by the presence of large velocity gradients; too fine and there are not enough protons to return adequate signal, thus necessitating longer scan times and/or thicker slices. These limitations can be largely overcome by contrast-enhanced (CE)-MRA, which resorts to the intravenous injection of a paramagnetic contrast agent to shorten the relaxation time of blood relative to background tissue. This results in a stronger blood signal within a shorter acquisition time, and minimises the saturation of slow flow. Like CTA, the intravenous injections must be timed carefully to avoid venous contamination. Unlike CTA, however, CE-MRA cannot yet be accomplished within a few seconds, and so pulse sequences must be designed carefully so as to minimise the effects of variable contrast agent concentration during the traversal of k-space.

In what amounts to the inverse of TOF-MRA, black blood (BB)-MRI seeks to suppress all signal, including that of blood, from *outside* the imaging slice or volume. As shown in Fig. 4.2 E, this makes it possible to see not only the (black) vessel lumen but also the (bright) vessel wall and surrounding tissue. As a result, BB-MRI – by definition imaging, not strictly angiography – is particularly attractive for characterising the vascular wall pathologies that can usually only be inferred from the presence of tortuosity or stenosis in traditional angiograms. In the context of vascular modelling, BB-MRI can be used to reconstruct both the lumen *and* wall simultaneously [480], which is particularly advantageous for providing outer wall (or simply wall thickness) boundary conditions for structural or fluid-structure interaction analyses (see Chapter 9). Like TOF-MRA, BB-MRI is based on the assumption that blood leaves the imaging region before it has time to recover its signal, so flow artifacts can also occur, albeit appearing as the inverse of those in TOF-MRA.

An alternative less susceptible to saturation/recovery effects is phase contrast (PC)-MRA, which relies on the fact that the velocity of moving spins can be encoded into the phase of the complex MR signal. Not only does this produce bright blood angiograms from the magnitude of the signal; as Fig. 4.2 F shows, the blood velocities themselves can be encoded into images of the signal phase, in principle one image for each direction of flow. By gating the acquisition to the cardiac rhythm, it becomes possible to produce time-resolved (cine) images of the blood velocities. In principle, this allows for the acquisition of a complete “4D” velocity field [43] ; in practise, such scans can be very time-consuming and so are not always tolerated by patients. PC-MRA tends not to be used for angiographic purposes alone, since for a given resolution and coverage, more scan time is required relative to TOF-MRA and especially CE-MRA.

4.1.3 Ultrasound (US) imaging

Relative to x-ray and MR imaging, US is the least expensive and least invasive. Similar to sonar and other echolocation techniques, US works by transmitting beams of low power, high frequency sound waves (on the order of a few MHz) into the body via a hand-held probe coupled to the skin through a conductive gel. Normally, this same probe receives the echoes caused by the strong reflections at tissue boundaries, and the weaker reflections within the tissues themselves. The distance between the probe and the reflections can then be calculated from the known speed of sound through tissue (1540 m/s) and the measured delay between transmission and each echo.

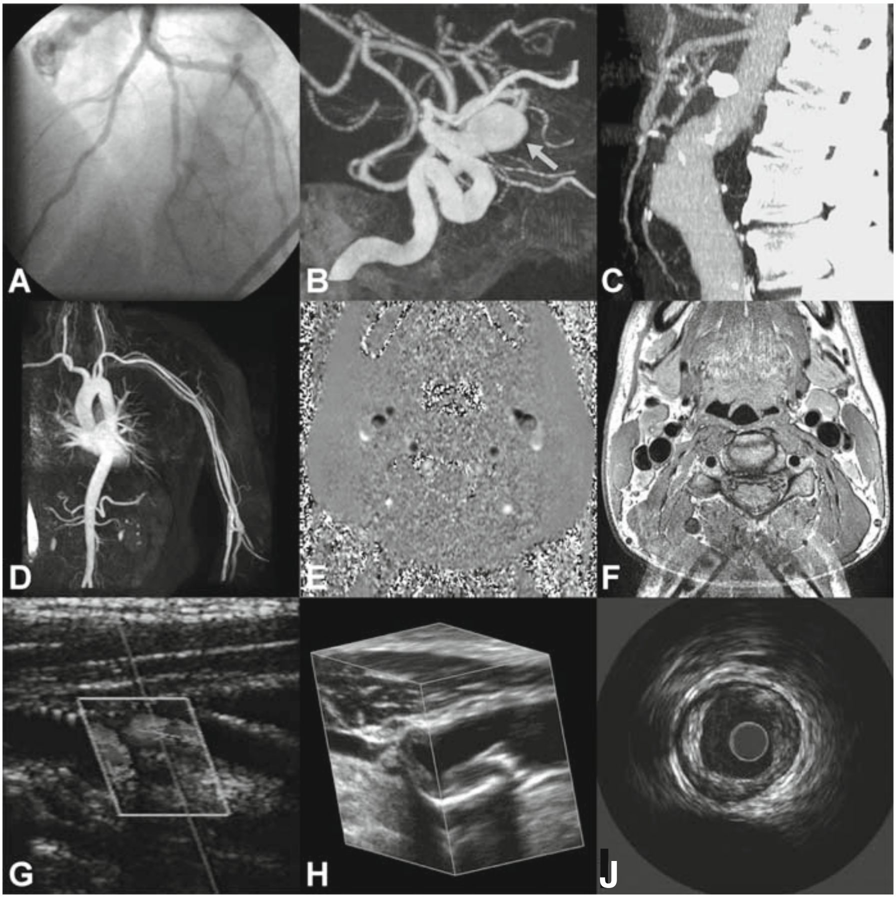

Fig. 4.2. Representative vascular images from the various modalities discussed. A. 2D x-ray angiogram of a coronary artery tree. B. Maximum intensity projection (MIP) through a 3D x-ray angiogram of a cerebral aneurysm (arrow). C. MIP through a 3D CTA of an abdominal aortic aneurysm. D. MIP through a 3D CE-MRA of the major arteries from neck to abdomen. E. 2D PC and F. BB-MRI images of the carotid arteries at the level of the bifurcation. G. Duplex US image of a carotid artery bifurcation. H. 3DUS image of a carotid artery bifurcation. J. IVUS image of a coronary artery in cross-section. Panels A&J courtesy of Dr. Jolanda Wentzel, Erasmus University. Panel D courtesy of Dr. Nils Planken, Department of Radiology, Amsterdam Medical Center. Panel G courtesy of Dr. David Spence, Robarts Research Institute. Panel H courtesy of Dr. Aaron Fenster, Robarts Research Institute

For typical depths of penetration, echo delays are less than a millisecond, and so US offers a remarkable facility for real-time imaging. For example, the amplitude of each echo can be assigned a grayscale value and displayed on screen at its respective depth from the probe. With the beam fixed this produces a M-mode image, so named because it sensitively depicts tissue motions at high ($\sim$ 1000 Hz) temporal resolution. More commonly, the US beam is swept, either mechanically or electronically, through a plane or sector to produce real-time ($\sim$ 100 Hz) 2D B-mode images, so named because their brightness reflects the distribution of tissue throughout the slice. By virtue of the Doppler effect, blood flow in the direction of the US beam produces shifts in the frequencies of the returned echoes, which can be converted into velocities and mapped onto a near-real-time ($\sim$ 15 Hz) colour Doppler image, as shown in the duplex US (i.e., combined B-mode and colour Doppler) image shown in Fig. 4.2 G.

An obvious limitation of conventional US imaging is the 2D nature of the acquisition, which, as noted earlier, makes visualisation of tortuous arteries difficult. (Remember that these are actual slices through the vessel, not projections.) This can be overcome via three-dimensional US (3DUS), which reconstructs an image volume from conventional 2D slices acquired during freehand or mechanical translation of the US probe (Fig. 4.2 H). Although 3DUS B-mode volumes can be acquired in less than a minute, longer scan times are required for 3D colour Doppler US images. Moreover, this approach is sensitive to vessel motion between individual 2D slices: 3DUS images can exhibit so-called "sawtooth" artifacts if the artery is pulsating as the probe is translated. These limitations, however, may soon be overcome through the development of 2D array transducers, which will permit real-time acquisition of 3D (and possibly 4D) US images.

As evidenced in Figs. 4.2 G and 4.2 H, ultrasound image quality and resolution are typically coarse relative to the other imaging modalities. Moreover, constructive and destructive interference of the scattered sound waves results in a form of multiplicative noise referred to as "speckle". As with MRI, image quality in US is also inextricably tied to the proximity of the vessel to the transducer at the surface: with increasing depth, less acoustic power is available for reflection. Similarly, tissue behind a calcification can be entirely masked because all of the acoustic power is reflected, leading to well-known shadowing artifacts. This also precludes the use of US for imaging cerebral vessels within the skull, except those visible through the thin temporal bone. Similarly, because the US also does not transmit well through air (hence the need for coupling gel between the transducer and the skin surface), imaging of abdominal vessels can be challenging. Image quality and artifacts cannot simply be overcome by increasing the power deposited, for this would risk heating the tissue. Instead, for deep or shadowed vessels like the coronaries, one resorts to intravascular ultrasound (IVUS), which places the transducer within the artery via a catheter (Fig. 4.2 J). Although this results in an invasiveness equal to that of x-ray angiography, it does provide information that angiography does not, namely, the surrounding tissue. As with BB-MRI, this is useful for providing the necessary boundaries for fluid-structure interaction analyses (see Chapter 9); however, because imaging planes follow the orientation of the (usually tortuous) vessel [261], adjacent slices are not parallel, potentially complicating the reconstruction of the 3D geometry (see Section 4.5).

4.2 Geometric modelling of blood vessels

Regardless the specific imaging modality, medical images contain information in the form of multidimensional arrays of numeric values, which are correlated with one or more physical properties of the imaged anatomical structures. Modelling the geometry of a vessel from medical images consists in extracting the location of the vessel wall from the values contained in the image volume. Since the image formation process in general alters the underlying anatomical information, and since the resolution with which the latter is acquired is bound to be finite, medical images are *coarse* representations of the underlying anatomy. Modelling the surface of an anatomical object from an image can therefore be regarded as an inverse problem.

More formally, a medical image can be seen as the scalar function $I : \Omega \rightarrow \mathbb{R}$ which associates an intensity value to every point in the acquisition volume $\Omega \subset \mathbb{R}^3$. In this framework, an image is not a collection of isolated samples, but rather a continuous function known at the nodes of a regular grid, whose values are interpolated over space with suitable shape functions (e.g. linear, cubic). On the basis of I, we are ultimately interested in finding a representation of the surface corresponding to the lumen wall boundary, or, in case the imaging technique allows it, as in the case of black blood MRI, to the inner and outer boundaries of the vessel wall. In both cases, we will generically indicate the obtained surface representations with $\mathbf{S}$, which will then serve as the definition of the boundary of the computational domain for the haemodynamic or fluid-structure interaction problem at hand, as introduced in Chapter 3.

Before we can face the problem of determining the location of $\mathbf{S}$ from I, it is necessary to describe how a generic 3D surface can be mathematically represented.

4.2.1 Explicit surface representations

A surface $\mathbf{S}$ corresponding to the vascular wall can be represented explicitly as a bivariate parametric function with values in $\mathbb{R}^3$ (as shown in Fig. 4.3, left). A common choice in computer graphics is expressing $\mathbf{S}$ as a set of adjacent polygons (e.g. triangles) whose vertexes have a known position. The surface in this case is given by

$$\mathbf{S} = \bigcup_i \mathbf{S}_i \tag{4.1}$$

with

$$\mathbf{S}_i(u,v) = \sum_{j=1}^{N_i} \phi_{i,j}(u,v)\, \mathbf{p}_{i,j} \tag{4.2}$$

where i indicates the i-th element, N_i is the number of nodes in the i-th element, $\phi_{i,j}$ is the shape function on element i, relative to node j, and $\mathbf{p}_{i,j}$

is the position of the j-th node in the i-th polygon. In general, the shape functions employed are linear Lagrangian basis functions satisfying

$$\phi_{i,j}(u_l, v_l) = 1 \qquad \text{for } l = j \tag{4.3}$$

$$\phi_{i,j}(u_l, v_l) = 0 \qquad \text{for } l \neq j \tag{4.4}$$

$$\sum_j \phi_{i,j}(u, v) = 1 \qquad \forall\ (u, v) \in \mathbf{U}_i \tag{4.5}$$

where U_i is the parametric space of the i-th element. In the case of triangular elements, suitable shape functions are barycentric coordinates. Due to the linear nature of the interpolation, a large number of small polygonal elements are required in order to represent a realistic anatomical structure. The representation does not offer any built-in control over the regularity of the surface apart from its continuity. However, given the simplicity of the formulation, it is very flexible with respect to the possible topologies that can be represented.

An alternative way of explicitly representing a surface is to use adjacent high-degree polynomial patches, such as NURBS (non-uniform rational B-splines) (as shown in Fig. 4.3, right) [403]. In this case, the surface takes the form of a union of patches each defined by a tensor product rule

$$\mathbf{S}_i(u, v) = \frac{\sum_{j=1}^{N} \sum_{k=1}^{M} \phi_{i,j}^{p}(u) \phi_{i,k}^{q}(v) w_{i,j,k} \mathbf{p}_{i,j,k}}{\sum_{j=1}^{N} \sum_{k=1}^{M} \phi_{i,j}^{p}(u) \phi_{i,k}^{q}(v) w_{i,j,k}} \tag{4.6}$$

where i indicates the i-th patch, $\mathbf{p}_{i,j,k}$ is one of the $M \times N$ control points, $w_{i,j,k}$ is the weight associated to the j,k-th point and $\phi_{i,j}^{p}$ a B-spline basis function of degree p, defined recursively as

$$\phi_{i,j}^{0}(u) = \begin{cases} 1 & u_j \leq u < u_{j+1} \\ 0 & \text{otherwise} \end{cases} \tag{4.7}$$

$$\phi_{i,j}^{p}(u) = \frac{u - u_j}{u_{j+p} - u_j}\, \phi_{i,j}^{p-1}(u) + \frac{u_{j+p+1} - u}{u_{j+p+1} - u_{j+1}}\, \phi_{i,j+1}^{p-1}(u) \tag{4.8}$$

where the values $u_1 \ldots u_{N+p+1}$ are called *knots* and satisfy $u_j \in [0, 1]$ and $u_{j-1} \leq u_j$. A realistic surface is obtained by employing several adjacent patches and expressing continuity conditions at their boundaries. Such conditions can reflect different degrees of continuity, namely positional continuity (C0), tangential continuity (C1) or curvature continuity (C2), which respectively require continuity of zeroth, first and second derivatives of adjacent parametric patches.

The degree p of the B-spline basis function controls the smoothness of the resulting surface and must be cautiously chosen in order to properly reproduce the desired surface features while avoiding high-degree artifacts, which appear as spurious oscillations typically located in the vicinity of patch boundaries. Compared to polygonal representations, an anatomically realistic surface can be represented by a lower number of NURBS patches and its smoothness, and

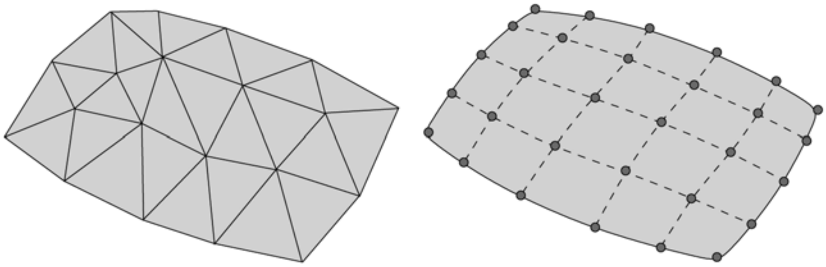

Fig. 4.3. Left, polygonal surface; right, NURBS surface (dashed lines represent isolines of the parameters u, v, while dots represent knots)

consequently the size of the smaller surface features, can be controlled more directly. However, the use of larger patches leads to reduced flexibility in the representable surface topology.

The advantages of the NURBS representation is the interpretation of the vertexes $p_{i,j}$ as a control polygon that permits an intuitive modification of the surface through changes in their position. Additionally, changes in the position of one of the vertexes affects the surface only locally, a desirable feature for CAD systems.

A representation that offers both topological flexibility and direct control over surface smoothness is constituted by *subdivision surfaces*. Since their description is not critical for the contents of this chapter, we refer the reader to specific publications on this matter [561].

4.2.2 Implicit surface representations

In addition to the explicit representation described so far, a surface can also be thought of as embedded in a function $F : \mathbb{R}^3 \to \mathbb{R}$, and represented by describing its embedding. For example, the surface $\mathbf{S}_s$ of a sphere of centre $\mathbf{c}$ and radius r is the locus of points $\mathbf{x} \in \mathbb{R}^3$ where the function $F_s(\mathbf{x}) = (\mathbf{x} - \mathbf{c})^2 - r^2$ is zero. In this case, the surface of the sphere is embedded in the 3D function F_s, that is, $\mathbf{S}_s$ is an isosurface (or the zero level-set) of F_s. The implicit representation of a surface $\mathbf{S}$ therefore becomes

$$\mathbf{S} = \left\{ \mathbf{x} \in \mathbb{R}^3 : F(\mathbf{x}) = k \right\} \tag{4.9}$$

where k is a value of choice.

For a realistic vascular surface, the function F is likely not known in closed form. Instead, it can either be given at the nodes of a regular grid or it can be defined as a superposition of analytical basis functions. The simplest example of the first case occurs when the surface of a vessel is assumed to coincide with a particular value of intensity on a clinical image. In this situation, the embedding function F is the image itself I, of which the vascular surface is an isosurface (as shown in Fig. 4.4). Embedding functions known at the nodes of

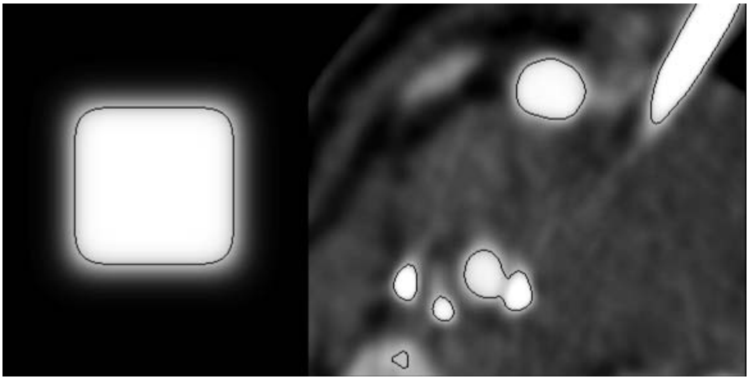

Fig. 4.4. Traces of embedding functions on single imaging planes, along with an explicit representation of one isolevel: left, analytic embedding function; right, medical image as an embedding function

a grid can also be obtained as the result of modelling procedures guided by image features, as shown later in the chapter. For example, in the case of *level sets* (see 4.4.4), the embedding function is generated as the finite differences solution of a PDE [453].

The embedding function is defined as the superposition of analytic functions in the case of *radial basis functions* (RBF), for which the value of a single radial basis function depends solely on the distance from a given location (hence their name). The embedding function in this case is given by

$$F(\mathbf{x}) = P(\mathbf{x}) + \sum_{i=1}^{N} w_i \, \phi(\|\mathbf{x} - \mathbf{x}_i\|) \tag{4.10}$$

where P is a low-degree polynomial function, w_i is a set of weights corresponding to the locations $\mathbf{x}_i$ and ϕ is the basis function. For modelling 3D shapes, radial basis functions take one of the following forms [375]

$$\phi(r) = \begin{cases} r & \text{linear} \\ r^3 & \text{cubic} \\ \exp(-Cr^2) & \text{Gaussian} \\ (r^2 + C^2)^{\frac{1}{2}} & \text{multiquadratic} \\ (r^2 + C^2)^{-\frac{1}{2}} & \text{inverse multiquadratic} \end{cases} \tag{4.11}$$

where $C > 0$ in all expressions. The choice among the various possibilities is performed on the basis of smoothness of the result and on the convergence properties of the numerical method employed to determine the coefficients w_i in Eq. 4.10, as shown in Section 4.5.3. Common choices for representing 3D surfaces have been the linear and the cubic RBFs, the former exhibiting better convergence properties, the latter resulting in a greater surface smoothness [67].

4.2.3 From implicit to explicit surface representations

Most numerical techniques for modelling hemodynamics require explicit geometric representations of the physical domain on which problems are solved. Therefore, we will here describe the operation that allows to obtain explicit representations from implicit ones. Contouring, or isosurface extraction, is the procedure by which an explicit representation of a surface **S** is generated from the 3D scalar function F that embeds it.

Marching cubes [299] is a widely used algorithm for generating polygonal explicit surface representations from embedding functions sampled on a regular grid. As already noted, medical images are embedding functions for their isosurfaces, so, provided the isolevel associated to the vascular wall is known (which is usually not true for real world images) Marching Cubes can be used to directly obtain the surface of a vessel (the limitations of this approach will be explained in Section 4.4.2).

The algorithm works on the basic idea that a polygonal surface whose vertexes lie on grid edges can intersect a cubical neighbourhood of the grid and partition its vertexes in a finite number of configurations, 64 reduced to 15 by symmetry, called topological cases.

Thanks to this property, the polygonal surface can be constructed, one cubical neighbourhood at a time, by selecting the topological case corresponding to the local partition of above and below vertexes. Compatibility of the surface from cube to cube is obtained owing to the fact that each couple of neighbouring cubes presents the same vertex partition on the shared face. After the selection of the proper topological case, the exact position of polygon vertexes is obtained by linearly interpolating the values of F along the intersected edges. Once each cubical neighbourhood has been visited, the polygonal explicit surface representation is generated (an example is shown in Fig. 4.5).

A potential problem with this algorithm is that, in a limited number of topological cases, two distinct polygon configurations can give rise to the same cube vertex partition. Since this could break surface continuity if incompatible choices were made for two neighbouring cubes, additional criteria have been proposed to select the case which guarantees topological consistency. This is an acceptable solution in most cases, when the surface to be reconstructed is regular. As an alternative, other methods have been proposed to solve the topological consistency problem, such as Marching Tetrahedra, in which space is subdivided in tetrahedra rather than in cubes, resulting in 16 possible polygonisations reduced to 3 by symmetry [299].

The previous algorithm can be applied whenever the function F is available at the vertexes of a regular grid. If the implicit representation is given analytically under the form of a radial basis function, a possibility for constructing an explicit surface is to sample the radial basis function over a regular grid and extract the isosurface of zero level using marching cubes. Alternatively, it is possible to exploit the analytic nature of F [42]. Instead of relying on

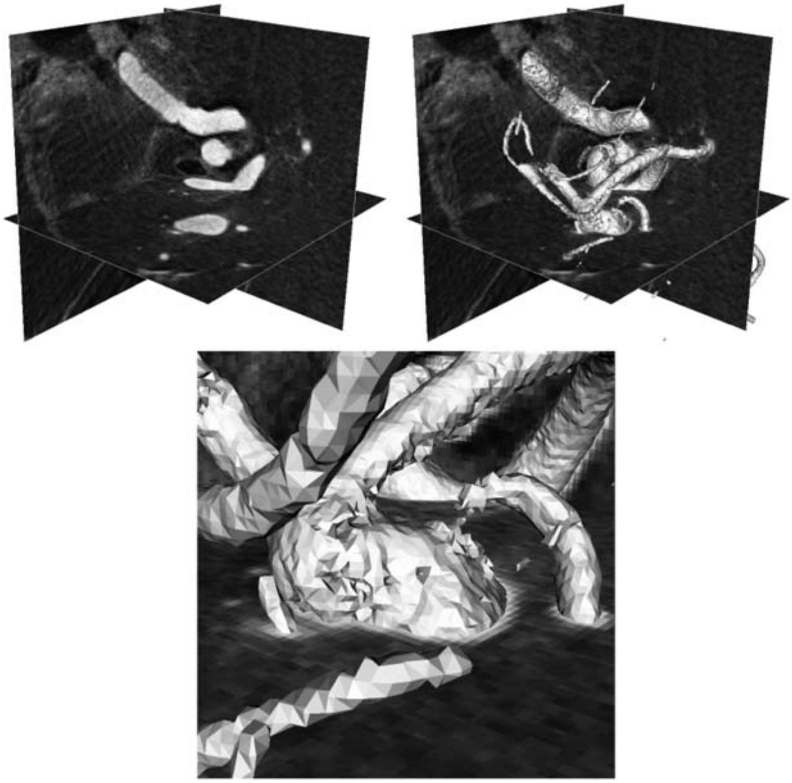

Fig. 4.5. Example of 3D contouring performed with the Marching Cubes algorithm on a CRA image of a cerebral aneurysm. Left, the 3D image; middle, the result of contouring at the level of the transition between the contrast-enhanced vascular lumen and the surrounding tissue; right, a detail showing the appearance of the resulting polygonal surface

a grid of predefined samples which are then linearly interpolated from grid point to grid point, the method takes advantage of the fact that values of F are available at all locations without interpolation, and accurately locates surface points by means of a root finding method in the vicinity of the desired contour value. Aside from this, the polygonisations method utilises similar concepts to those used in marching cubes and marching tetrahedra.

4.3 Image enhancement

In dealing with the problem of building representations of the geometry of blood vessels, we have assumed that the location of their surface was known. We will now start to face the problem of how to actually derive such information from medical images.

Medical images can be affected by noise and artifacts that may interfere with the segmentation process, or the image data may be provided in a form not suitable for the segmentation method of choice, as in the case of anisotropic resolutions, for which the spacing of the imaging grid along different axes is different. In order to alleviate these problems, an imaging enhancement step can be performed prior to segmentation. The introduction of such a step can substantially alter the information contained in an image and potentially play a role in the outcome of haemodynamic modelling. Since such influence is associated in general with a set of user-dependent parameters, care must be taken at defining the minimum amount of image enhancement operations needed for segmentation to be successful.

4.3.1 Resampling

A first pre-segmentation operation we describe is image resampling. Its goal is to change the resolution of the image in one or more imaging axis directions. For modality-dependent reasons, images can be acquired with resolutions that do not match the requirements imposed by the segmentation method of choice. This leads to the need of changing (in most cases, increasing) the spacing of the imaging grid in one or more directions, granted that this operation will in no case increase the information content of the image.

Image resampling therefore consists in projecting image intensity values onto a different grid than the original one. The most common way of performing resampling is to obtain the image values at the new grid points by interpolation from the original imaging grid. The choice of the interpolation operator affects the accuracy of the resampling. If subsampling is needed, interpolation can be preceded by low-pass filtering (see next section) of the original image in order to avoid the generation of artifacts.

Among the available interpolation methods, the most common are nearest neighbour and linear interpolation, which correspond to constant and first order shape functions over the imaging grid. More sophisticated interpolation

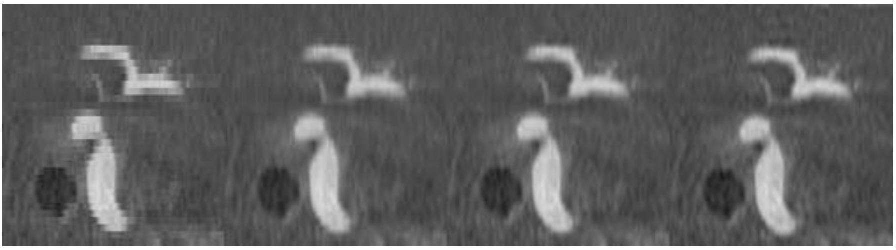

Fig. 4.6. From left to right, CRA image with anisotropic resolution (the number of points along the horizontal direction is four times greater than along the vertical direction), resampled with nearest neighbour, linear, cubic and sinc interpolation

methods include B-spline [514] and windowed sinc interpolation. In the former, a representation of the image based on bell-shaped high-order (typically cubic) polynomials centred at the imaging grid points is first obtained by least squares approximation. The image is then resampled onto the new grid by linear combination of the polynomial basis functions. In the latter, off-grid values are obtained by convolution of the image with a cardinal sine kernel. Convolution of the image I with a kernel K is defined as

$$I * K(\mathbf{x}) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I(\mathbf{x})\; K(\mathbf{x} - \bar{\mathbf{x}})\; d\bar{\mathbf{x}}, \tag{4.12}$$

where in this case the kernel is defined as

$$K(\mathbf{x}) = \mathrm{sinc}(\mathbf{x}) = \frac{\sin \mathbf{x}}{\mathbf{x}}. \tag{4.13}$$

For the convolution kernel to have finite support, the cardinal sine function is restricted to a suitable window, hence the name. Since the Fourier transform of the sinc function is the box function, and since the convolution of two functions corresponds to a product in frequency space, this interpolation method corresponds to extending the periphery of the Fourier transform of the original image by a zero-padded region. Although theoretically optimal in terms of Shannon's sampling theory[1], the latter interpolation method can give rise to ringing artifacts around sharp edges. A sample evaluation of the described image interpolation method is shown in Fig. 4.6.

A more sophisticated interpolation method is *adaptive control grid interpolation* [166], which borrows concepts from motion estimation and frame prediction algorithms and aims at reconstructing missing inter-slice information for datasets suffering from highly anisotropic resolutions.

4.3.2 Noise reduction

Noise is generally identified as a random high-frequency signal added or multiplied to the image content. Noise is inherent in any acquisition process and it is ascribable to thermal noise in the signal processing electronics or to physical undesired signal sources. Reduction of noise can be obtained either by means of a smoothing filter, which removes the high-frequency components of the image, or by modelling noise characteristics and filtering them out. In the first case, no prior knowledge about the noise generation process is required, but, on the other hand, meaningful high-frequency image content can be filtered together with noise. In the second case, noise reduction is more specific, even if this depends on the accuracy of noise modelling. The first approach is

[1] The Shannon sampling theorem is a well-known result in information theory. Conversion of a continuous function into a numeric sequence is called *sampling.* Shannon theorem (or Nyquist-Shannon theorem) states that a continuous function featuring the highest frequency ν_M is completely determined by a sampling with frequency $2\nu_M$.

the most widely used in vascular modelling from CT and MR images, probably because noise for these modalities has simpler characteristics and signal-to-noise ratio (SNR) is usually fairly high. Moreover, the typical scale of the anatomical structures of interest in CT or MR is often much larger than that of noise. In contrast, more specific approaches are used for B-mode ultrasound image enhancement, which are affected by speckle noise, already introduced in Sect 4.1, whose scale is potentially similar to that of the observed structures.

One of the simplest and most used forms of noise reduction in images is Gaussian filtering. It consists in performing a convolution of the image $I(\mathbf{x})$ with a Gaussian kernel

$$G_\sigma(\mathbf{x}) = \frac{1}{\sqrt{(2\pi\sigma^2)^3}} e^{-\frac{\mathbf{x}^2}{2\sigma^2}} \tag{4.14}$$

where σ is the width of the Gaussian kernel and controls the amount of smoothing, i.e. the maximum spatial frequency represented in the filtered image. An equivalent approach is the simulation of a diffusion process for image intensity values governed by a PDE of the form

$$\frac{\partial I(\mathbf{x},t)}{\partial t} = c\,\Delta I(\mathbf{x},t) \tag{4.15}$$

where c is the diffusivity coefficient which regulates the strength of diffusion. We here underline that this formulation is made possible by the fact that the image is considered as a continuum of intensity values rather than a collection of isolated samples. The PDE is then solved numerically, for instance by using finite differences (see Chapter 2), over the imaging grid up to a time $\bar{t}$. Solving the diffusion equation with c and $\bar{t}$ is analogous to performing a convolution with a Gaussian kernel having $\sigma^2 = 2\,c\,\bar{t}$.

The drawback of the previous approaches is the potential loss of information that results from filtering an image in which signal and noise have similar spatial scales. In particular, objects in images are ideally separated from the background by sharp borders, which have a high frequency content. Smoothing can therefore result in edge blurring and displacement, which can affect both the estimation of surface geometry and alter the topology of the resulting object. In order to limit the effect of smoothing on relevant edges, anisotropic diffusion filtering has been introduced. The idea behind this approach is that smoothing should be stronger in regions dominated by noise, while it should preserve regions in which image features, such as object boundaries, are predominant. Obviously, this would require the definition of object boundaries, which in turn is our ultimate goal. Instead, image features are estimated from the gradient magnitude of the intensity values or by more sophisticated edge-detection techniques. In the standard formulation [384], the anisotropic diffusion equation is given by the following PDE

$$\frac{\partial I}{\partial t} = \nabla \cdot c(|\nabla I|)\,\nabla I \tag{4.16}$$

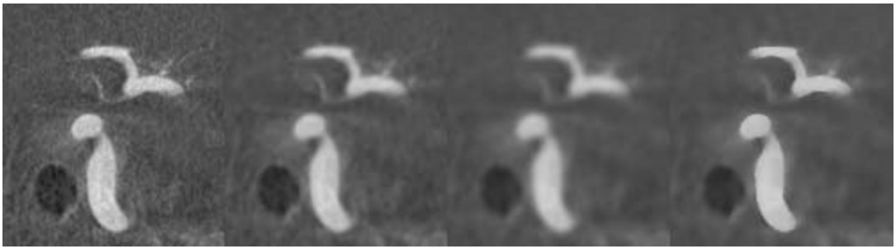

Fig. 4.7. From left to right, CRA image affected by noise, convolved with a Gaussian kernel with $\sigma = 2.5$ pixels, with a Gaussian kernel with $\sigma = 5$ pixels and filtered with anisotropic diffusion with the equivalent of $\sigma = 5$ pixels

where the diffusivity can be expressed as a decreasing function of the image gradient magnitude, such as

$$c(\mathbf{x}, t) = e^{-\left(\frac{|\nabla I|}{k}\right)^2} \tag{4.17}$$

or

$$c(\mathbf{x}, t) = \frac{k^2}{(k^2 + |\nabla I|^2)} \tag{4.18}$$

where k is used to control the relative influence of image features over the diffusion process [384]. This way, the diffusion process preserves those regions at which intensity values vary more rapidly, such as at object edges. The filter must be tuned for specific applications in terms of k and $\bar{t}$. Since the interpretation of the parameters is less straightforward than in the linear case, empirical evaluation of the effects of the filter is necessary.

An interesting variation of the classical anisotropic diffusion equation above is constituted by the *modified curvature diffusion equation* (MCDE) [541]

$$\frac{\partial I}{\partial t} = \nabla \cdot \left(c \, \frac{\nabla I}{|\nabla I|} \right) |\nabla I| \tag{4.19}$$

where diffusivity can take the same form as in the previous case. This equation, which is regarded as the evolution of a *level set* (see Section 4.4.4) analogous to anisotropic diffusion, can be interpreted as a feature-dependent flow of intensity isosurfaces controlled by their curvature, represented by the term $\nabla \cdot \frac{\nabla I}{|\nabla I|}$. This results in a flattening effect on high-curvature image isosurfaces, such as the ones produced by small-scale noise. This filter does not exhibit a feature edge enhancement effect as in the previous case. A sample evaluation of the described noise reduction methods is shown in Fig. 4.7.

4.3.3 Vessel enhancement

While the techniques presented so far can be applied to images independently from their specific content, *a priori* assumptions on the expected appearance of the structures of interest allow to formulate methods targeted at highlighting those structures only, such as *multiscale vessel enhancement* [167]. The aim of this method is to enhance vessel-shaped structures with respect to the rest of the anatomical structures based on the analysis of the eigenvalues of the Hessian matrix $\mathcal{H}$, of the image intensity.

Let's first recall the definition of Hessian matrix of an image at one point,

$$\mathcal{H}_\sigma = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \tag{4.20}$$

where

$$I_{\alpha\beta}(\mathbf{x}) = \sigma^2 \frac{\partial^2 G_\sigma(\mathbf{x})}{\partial\alpha\,\partial\beta} * I(\mathbf{x}) \tag{4.21}$$

that is, the convolution of the image with the second derivative of the Gaussian kernel with width σ. This corresponds to a convolution of $I(\mathbf{x})$ with $G_\sigma(\mathbf{x})$ and taking the second derivatives of the resulting image, which selects from the image the spatial scales above σ.

The basic idea behind the filtering technique is that a tubular structure should be characterised by low intensity variations along the axis of the vessel and by a certain degree of cross-sectional symmetry. In terms of Hessian matrix eigenvalues, denoted as λ_1, λ_2, λ_3 and sorted so that $|\lambda_1| \leq |\lambda_2| \leq |\lambda_3|$, the idea translates into the fact that, at points inside tubular structures, λ_1 should be low, and λ_2 and λ_3 should be high and of equal sign (negative for bright intensity vessels and positive for dark intensity vessels).

Based on these considerations, vessel enhancement for bright intensity vessels of scale σ can be obtained as the output of the following function

$$\mathcal{V}_\sigma(\mathbf{x}) = \begin{cases} 0 \quad \text{if}\, \lambda_2 > 0 \,\text{or}\, \lambda_1 > 0, \\ \left(1 - e^{-\frac{\mathcal{R}_A^2}{2\alpha^2}}\right) e^{-\frac{\mathcal{R}_B^2}{2\alpha^2}} \left(1 - e^{-\frac{\mathcal{S}^2}{2c^2}}\right) \quad \text{otherwise} \end{cases} \tag{4.22}$$

where $\mathcal{R}_A = \frac{|\lambda_2|}{|\lambda_3|}$, $\mathcal{R}_B = \frac{|\lambda_1|}{\sqrt{|\lambda_2\lambda_3|}}$ and $\mathcal{S} = |\mathcal{H}_\sigma| = \sqrt{\sum_j \lambda_j^2}$. Parameters α, β and c are scaling factors. Since Equation (4.22) depends on the scale σ at which second derivatives are computed, the filter is applied at multiple scales and the optimal scale is selected as the one which yields the maximum filter response, that is $V(\mathbf{x}) = \max_\sigma \mathcal{V}_\sigma(\mathbf{x})$.

Besides being used for highlighting vessels in image datasets, the filter has been successfully employed for automated centreline detection [167], stenosis grading [168, 520], and level sets initialisation for segmentation [119]. Indeed, although effective for vessel detection, the filter does not ensure that the shape of the lumen is not altered, therefore it is more suited in initial phases of vessel identification than in precise location of the vessel wall.

Lastly, it must be noted that this image enhancement approach relies on assumptions that may not be verified for all imaging modalities and vascular segments. For example, the presence of non tubular vascular geometries, like aneurysms, complex bifurcations or calcifications, which in CT images appear brighter than the contrast medium, may give rise to low filter output inside vessels. Moreover, in black blood MRI imaging, the lumen appears dark, the wall is in general bright and the surrounding tissue may again appear dark, which does not lead to cylindrical structures but rather to thick walled hollow ones, which does not comply with the method assumptions.

4.4 Image segmentation for vascular modelling

The task of deriving the shape of an object from a digital image is commonly referred to as image segmentation. In its most classical acception, image segmentation implies that the pixels constituting an image are partitioned into classes corresponding to the objects represented in the image and to the background. In our context, however, the identification of the boundary of a vessel needs to be carried out at sub-pixel precision, for the resulting model to be suitable as the physical domain of a CFD simulation. In this sense, the automatic identification of vessels in an angiographic image is not as important as the accurate determination of the location of the lumen boundary. In fact, while a user can robustly identify the *presence* of a vessel on an image, a *precise* definition of its boundary is a challenging task for an operator in the absence of objective criteria, the implications of which are discussed in the next paragraph. Since the reconstruction of the vessel shape has a great influence in the modelled hemodynamics, as it will be clear in Chapter 5, we will focus on those segmentation techniques geared at accurately and precisely estimating the shape of the lumen rather than those designed for automatically identifying the presence of vessels in an image.

4.4.1 Manual segmentation

Manual placement of points or delineation of contours on single image planes is the first form of segmentation we take into account. This approach is based upon the decisions of an operator who identifies the boundaries of the structures of interest based on visual perception of image content. Obviously, this method is affected by a potentially low reproducibility of the results, especially when the image is of difficult interpretation. On the other hand, for images heavily affected by artifacts of low signal-to-noise ratios this might be the only option, as automated methods may not perform as well as human perception in filling missing information, or, if they did, they may require a large number of user-defined parameters that practically make segmentation completely operator dependent, only in a less intuitive way.

While the use of corrupted images for hemodynamics modelling is questionable regardless of whether manual or automated segmentation is performed, in most cases user-driven point placement or contour delineation is useful as an initial approximation for automated segmentation methods shown in the following sections. For several of such methods, in fact, the closer the initial segmentation is to the target shape, the more straightforward is the segmentation process, although a good segmentation method should be reasonably robust to initialisation.

In general, although operator dependency has to be minimised, completely automated segmentation of real-world medical images is arguably not a strict requirement for haemodynamic modelling, at least not as the requirement of locating the position of the vessel wall as accurately and reproducibly as possible. Rather, this is achieved by adopting a segmentation method based on mathematical methods, while allowing the user to steer the segmentation process, therefore integrating human perception with objective image-based criteria.

4.4.2 Thresholding and classification

The term thresholding refers to the identification of one or more image intensity values that separate the anatomical structures of interest from the background. The determination of thresholds can be performed manually or algorithmically, and it can take place once for the whole 3D image or on a slice-by-slice basis. The latter approach is necessary when the signal is not uniform among the slices, such as in TOF-MRA, in which the signal depends on the amount of flowing blood, or in CTA, in which the density of the contrast agent can vary during the acquisition, resulting in a change in intraluminal signal along the z-axis. Once one or more thresholds have been selected, thresholding can either proceed by generating a classified image containing discrete labels based on the relative positions of the original image intensities with respect to the threshold set, or, in case only one threshold has been defined, by performing contouring on the image at the threshold isolevel and generating an explicit representation of the corresponding surface.

Since clinical imaging techniques do not associate a precise intensity value to a particular tissue class, manual thresholding can present several shortcomings if geometry has a great impact on the subsequent analysis, as in the case of vascular modelling. The choice of a threshold intensity level strongly affects the size and the topology of the resulting surface, since the vascular lumen is characterised by strong spatial variations of image intensity compared to uniform tissue regions, and therefore the sensitivity of the resulting surface to the chosen threshold is high.

It is therefore of crucial importance to identify objective criteria capable to define thresholds corresponding to the vascular lumen boundaries. One of them, and the most used in the clinical setting, is the *full width at half maximum* (FWHM) criterion, which identifies the lumen boundary at an intensity level, I_{FWHM}, halfway between the peak intensity within the lumen and the intensity of the background (see Fig. 4.8). This criterion is mostly used for measuring vessel diameters by plotting image intensity along a line that crosses a vessel and measuring the distance between points at I_{FWHM} on both sides of

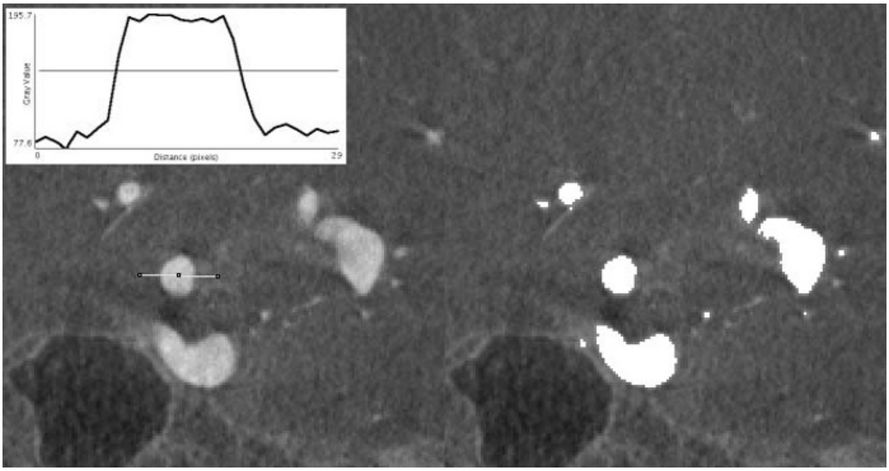

Fig. 4.8. Full width at half maximum (FWHM) criterion applied to a CRA image. Image intensities along the traced line are plotted at the upper left corner. On the right, the image is thresholded at the FWHM value

the lumen. This method has been proven to yield accurate estimates of diameters even for small vessels, for example, in TOF-MRA, while in PC-MRA the *full width at 10 %* criterion has been shown to be more accurate [225]. However, the FWHM criterion is sensitive to the definition of the peak lumen intensity, which could change with the location and the vessel size, and of the background, which in clinical images might consist of anatomical structures of heterogeneous intensities.

In contrast to the FWHM criterion, which is a local criterion, statistical thresholding methods seek to determine optimal thresholds from the distribution of intensity values over the whole image. A popular choice among statistical thresholding methods is Otsu's algorithm [362]. The method seeks to partition the image intensity histogram into classes in such a way that the between-class variance is maximised, where the between-class variance is defined as

$$\sigma_b^2 = \sum_{i=1}^{N_c} \omega_i (\mu_i - \mu)^2 \tag{4.23}$$

where N_c is the number of classes, ω_i is the probability of a pixel of belonging to the i-th class, μ_i is the mean intensity of the i-th class and μ the overall image intensity. This method has been employed in [375] to segment vessels from TOF-MRA images.

Several other statistical criteria can be used for determining thresholds that respond to different optimality criteria, such as K-means and expectation maximisation for the estimation of mixture models, to cite a few [386].

Available statistical methods for pixel classification differ widely from pure thresholding methods, and assign each pixel to a particular anatomical tissue class based on other criteria, such as adjacency relationships, Markov random fields, or prior information about shape [386]. The vast amount of literature on this topic cannot be covered here. It only has to be stressed that most classification criteria work at the pixel level, so that the sub-pixel positioning of the vessel boundary must be sought in a post-classification stage, following smoothness criteria or optimising the position of the surface on the basis of image features, as with deformable models, described later on. In this sense, classification methods can also be employed as automatic initialisation for deformable models.

4.4.3 Region growing and front propagation

While classification methods typically produce a segmentation of the whole image, the interest in vascular modelling is typically to segment vascular structures. Region growing methods generate selective segmentations starting from a pre-segmented region (or single pixel) and iteratively add neighbouring pixels to the region if they satisfy specified homogeneity criteria. The procedure stops when no more pixels can be added to the region. The initialisation is usually performed interactively, and the resulting segmentation clearly depends on the homogeneity criterion adopted. Such criterion is commonly related to the intensity of the pixel under scrutiny with respect to the pixels belonging to the segmented region. The simplest criterion consists of accepting the pixel if its intensity lies between a pair of user-defined thresholds. A more conservative criterion states that a pixel is included if its intensity and that of its neighbours fall between user-defined thresholds. In both these cases, the criterion does not vary throughout the growth. In a more sophisticated approach, the pixel is accepted if its probability of belonging to the region is above a certain threshold. For instance, under the assumption that the intensities of the pixels belonging to the segmented structure are normally distributed around a central value, a pixel is accepted in the region if its intensity lies in the interval $[\mu - f\sigma, \mu + f\sigma]$, where μ and σ are the mean and standard deviation of the pixel intensity in the segmented region and f a user-defined factor expressing the permissiveness of the criterion [241]. Since they consist in adding individual pixels to the segmented region, region growing methods are not capable of delivering sub-pixel precision. However, the final shape can eventually be smoothed (which does not ensure sub-pixel precision, but at least produces a realistic surface) and they can also be fruitfully employed as initialisation tools for deformable models.

A conceptually related approach to region growing is represented by front propagation methods. This approach is based on tracking the propagation of a wavefront from a seed point over the image. The speed of the wave is regulated by image features, and is typically set to be lower in regions where the image intensity varies more rapidly and higher where the image is uniform. The purpose is to make the front move rapidly towards the vessels while slowing down as it approaches vessel boundaries. The most popular front propagation algorithm is represented by the fast marching method, which provides an efficient solution to the eikonal equation

$$\|\nabla T(\mathbf{x})\| = F(\mathbf{x}) \qquad \text{with } F(\mathbf{x}) > 0 \tag{4.24}$$

where $T(\mathbf{x})$ represents the first arrival time of the wave at a point of the domain and $F(\mathbf{x})$ is the reciprocal of the local wavefront speed. In other words, we are looking for the spatial distribution of the arrival times of a wave of given local speed. Once the initial front and $F(\mathbf{x})$ are defined, the eikonal equation is solved on the image grid using upwind finite differences. A more efficient solution of the eikonal equation can be achieved by only tracking the solution in a narrow band of pixels defined around the front. Once computed, the isocontours of $T(\mathbf{x})$ represent a collection of surfaces describing the shape of the waveform during the propagation. It is the user's responsibility to select the value of $T(\mathbf{x})$ that captures the desired surface. Since it produces a continuous function, this front propagation method is able to deliver sub-pixel precision. In practise, though, the fact that the front only slows down without actually stopping on the vessels boundary makes the use of this method alone not particularly robust for segmentation, since the shape of the segmented vessel will depend on the choice of the contouring value of $T(\mathbf{x})$. In other words, the method lacks an objective criterion for defining the location of the boundary of vessels based on image intensity. Nevertheless, this method is very effective for the initialisation of deformable models, which will be the subject of the next section.

4.4.4 Deformable models

Deformable models are curves or surfaces defined in the image space which change their shape on the basis of image features, external constraints and internal deformation laws [319]. This way, the position of a curve or a surface can be optimised to identify the position of lumen boundary with sub-pixel precision.

Deformable models are not conceptually different from other segmentation methods: they serve to identify the position of the boundary of the shape of interest from the information contained in an image. Their peculiarity lies in the fact that the segmentation problem is treated as an energy minimisation problem, where the energy depends on the position and shape of the curve or surface and on image features. The fact that the minimisation process proceeds through deformation of the model towards its final shape presents two main advantages: the evolution can be followed and eventually steered in an intuitive way, and the initialisation can be performed by providing a good geometric approximation to the final shape. Being an optimisation process, segmentation with deformable models makes it possible to establish a trade-off between local adherence to image features and global criteria (e.g. regularity of the resulting shape), thus allowing a control over the single contributions to the final result.

From a formal point of view, the energy functional can be expressed as the sum of an internal and an external energy

$$\mathcal{E} = \mathcal{E}_{int} + \mathcal{E}_{ext}. \tag{4.25}$$

The internal energy term governs how the model changes its shape, for example it can confer a membrane or thin plate-like behaviour to the model, or control its regularity in terms of smoothness. External energy terms include image-based terms and positional constraints. The former drive the deformation based on the features the underlying image, for example inflating the model with different speeds according to image intensity, or attracting the model towards edges, while the latter serve to steer the evolution of the model with user-defined rules, e.g. to avoid specific regions of the image in the presence of artifacts or to help the convergence of the model to the features of interest.

Deformable models can be employed both in 2D, i.e. deformable contours on planar images, or in 3D, i.e. deformable surfaces in a 3D image, depending on the nature of the data (disjoint images or thick slice imaging typically require 2D deformable models) and the geometry of the structures to reconstruct (vascular networks of complex geometry are more challenging to segment in sequences of 2D sections). Furthermore, the choice is also dictated by the need of user interaction for steering the evolution of the deformable model, e.g. in order to avoid the erroneous segmentation of flow-related artifacts arising in black blood MRI. Since interaction with a contour on a 2D slice is simpler than that with a surface within a 3D volume, 2D segmentation is often preferred in this case.

There are two major classes of deformable models, which correspond to the way the curve or surface is represented, namely *explicit* and *implicit* deformable models. Recalling the distinction made in Section 4.2, a surface $\mathbf{S}$ can be given by explicitly describing the position of its points or by describing an embedding function the surface is an isosurface of. In this perspective, the evolution of a deformable model can either be represented by explicitly tracking the position of its points, of by tracking changes in the embedding function. In the next two sections, the two approaches, along with their strengths and limitations, will be described independently from the specific form that $\mathcal{E}_{int}$ and $\mathcal{E}_{ext}$ can take. Subsequently, a description of image-based and internal forces applicable for both deformable model approaches will be provided.

Explicit deformable models

For explicit deformable models the position of surface points is explicitly represented and tracked during the evolution. In this case, the minimisation of Equation (4.25) acts on $\mathbf{S}(\mathbf{u})$, that is, on the way points in the parametric space are mapped onto the image space.

Two popular versions of explicit deformable models in 2D are *discrete dynamic contours* (DDC) and deformable splines, both commonly referred to as *snakes.* DDC are polylines, i.e. nodes connected by straight segments, whose deformation is expressed in terms of the change in the nodal coordinates. Deformable splines are instead piecewise polynomials with continuity constraints, each polynomial piece described by a set of control points. DDC are in general more flexible in terms of topology and easier to handle from a computational point of view, being described by nodal coordinates expressed in physical space. On the other hand, splines require fewer control points and implicitly incorporate internal constraints on their smoothness during the evolution. The choice between the two is in general application-dependent.

The general expression of the internal energy of a 2D snake is

$$\mathcal{E}_{int} = \int (w_1 |\frac{\partial \mathbf{S}}{\partial u}|^2 + w_2 |\frac{\partial^2 \mathbf{S}}{\partial u^2}|^2) du, \tag{4.26}$$

where u is the arclength, the first and second terms in the integral represent the snake's tension and stiffness, respectively. The relative contributions of the two terms are controlled by the weights w_1, w_2, chosen by the user on the basis of image characteristics. The external energy is in general expressed through the definition of a image-dependent potential

$$\mathcal{E}_{ext} = \int w_3 P(\mathbf{S}) du. \tag{4.27}$$

More details on the choice of P will be given in Section 4.4.4. An example of snake evolution is shown in Fig. 4.9.

The minimisation of $\mathcal{E}$ in Equation (4.25) can be worked out either directly in its energetic form, e.g using a gradient descent minimisation approach on $\mathcal{E}$, or in its local form, by directly expressing the problem in terms of the evolution of the position of physical points of $\mathbf{S}$ with respect to u, leading to the evolution equation

$$\frac{\partial \mathbf{S}}{\partial t} = w_1 \frac{\partial^2 \mathbf{S}}{\partial u^2} - w_2 . \frac{\partial^4 \mathbf{S}}{\partial u^4} - w_3 \nabla P(\mathbf{S}). \tag{4.28}$$

For DDC, a simple way to discretise equation 4.28 is by means of finite differences in the form of a *spring-mass* analogy, by which nodes are assigned a mass m, and the linear segments linking adjacent nodes are associated a stiffness. The nodal evolution equation for DDC is then expressed as [266]

$$m\, \mathbf{a} = w_{ext}\, \mathbf{f}_{ext} + w_{int}\, \mathbf{f}_{int} + w_d\, \mathbf{x}' \tag{4.29}$$

where $\mathbf{a}$ is the acceleration of nodes, $\mathbf{f}_{ext}$ and $\mathbf{f}_{int}$ are the external and internal forces applied to them, and the last term represents damping, introduced in order to avoid the set up of oscillations during the DCC evolution. The weights w_{ext}, w_{int} and w_d are again user-defined parameters.

The 3D counterparts of 2D snakes, commonly regarded as *balloons*, comprise deformable meshes and deformable spline surfaces. The considerations for the 3D case as well as the mathematical treatment of the energy terms

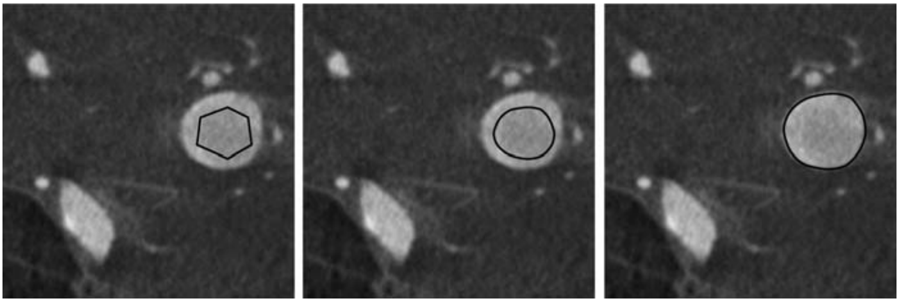

Fig. 4.9. Evolution of a 2D snake. Left, initialisation by interactive definition of the vertexes of a polygonal line; middle, refinement (or spline interpolation) under a smoothness constraint; right, deformation under the action of external image-based forces

and the minimisation problem are not substantially different from the 2D case. The parameter space is now represented by a bidimensional vector (u, v), and the internal energy terms represent local surface stretching and bending. As in the 2D case, the evolution of deformable meshes has been implemented resorting to the spring-mass analogy, although more sophisticated approaches based on the same idea assign a stiffness to the triangular mesh elements considered as solid surface elements rather than to their edges only [97, 318]. A typical evolution equation for an explicit 3D deformable model is

$$\frac{\partial \mathbf{S}}{\partial t} = w_1 G(\mathbf{S})\,\mathbf{N} + w_2 \Delta \mathbf{S} - w_3 \nabla P(\mathbf{S}) \tag{4.30}$$

in which the first term models inflation, by which the surface deforms along its normals with speed $G(\mathbf{S})$, the second term represents rigidity and the third term the action of an external force expressed as the gradient of a potential P, in general function of image features.

Specialised deformable models have been proposed for segmentation of vascular structures [167, 551]. In particular, Frangi et al [167] proposed the use of tensor-product B-splines in which one dimension is periodic and whose control points are constrained to move radially with respect to the estimated centreline, so that the deformable model maintains a cylindrical shape throughout the evolution. Similarly, but improving the behaviour of the model to bending, Yim et al [551] presented a *tubular deformable model*, consisting in a deformable mesh whose surface points are described in terms of radial distance and angle around the vessel axis, and are constrained to deform along radial lines. The main strength of these approaches is also their limitation, in that non-tubular vascular structures, such as complex bifurcations or aneurysms, are not easily reconstructed.

The main difficulty when dealing with explicit deformable models is the fact that they rely on specific parametric representations of $\mathbf{S}(\mathbf{u})$. In principle, the parametric space is fixed during the evolution; only the mapping from

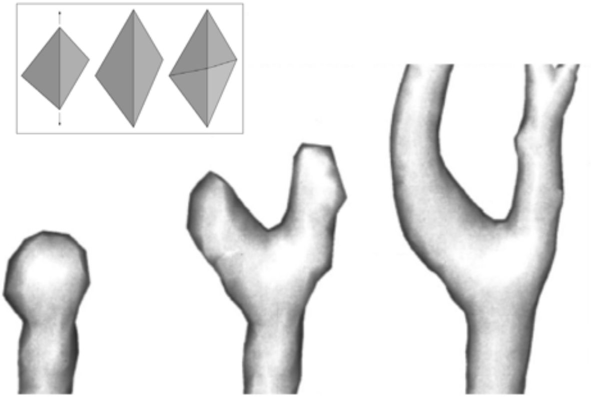

Fig. 4.10. Evolution of a polygonal balloon inflating under image-dependent forces. The surface is re-parametrised during the evolution to account for large deformations: if a triangle has one edges exceeding a threshold length, the two triangles sharing the edge are replaced by four smaller triangles (shown in the box)

the parametric space to the physical space is updated. In other words, a triangle in a polygonal balloon is allowed to have its nodal coordinates, but not its connectivity relationships with neighbouring triangles, updated during evolution. When the deformable model evolves, large deformations can occur, and the mapping from parametric to physical space can degenerate, or not be sufficiently refined to properly represent the underlying geometry. This issue is overcome with the introduction of re-parametrisation criteria, such as those employed by Ladak et al [265], which allow the mesh to be dynamically refined as the model undergoes large deformations. In particular, if the length of a triangle edge exceeds a user-defined threshold, the edge is split in two by the insertion of an additional node, and the two triangles sharing the edge are replaced by four triangles sharing the inserted node, as shown in Fig. 4.10 (see also Section 4.11.4).

The need for re-parametrisation is not only linked to large deformations. The initial topology of the deformable model may not correspond to that of the target shape. As an example, when an initially spherical model enters into two separate branches that merge back together, the model topology must change from that of a sphere to that of a torus, requiring an intersection detection phase followed by a global re-parameterisation. The problem of merging triangulated surface meshes has been addressed by Cebral et al [71] as a post-processing step. Topology-adaptive deformable models have also been proposed [320], for which the position of the deformable model is tracked on a background regular grid made up of simplexes (triangles in 2D and tetrahedra in 3D). Collision detection is performed by keeping track of the intersections of the surface on the grid; topology changes are then obtained by re-triangulating the surface from the grid once self-intersections have been removed.

Implicit deformable models

As discussed previously, a surface $\mathbf{S}$ can be represented implicitly as the isosurface of a function F defined over $\mathbb{R}^3$ with values in $\mathbb{R}$. Therefore, the description of an implicitly represented deformable surface is given by tracking changes in the embedding function. The resulting implicit deformable model is commonly referred to as level set [453].

The link between the deformation of $\mathbf{S}$ and the evolution of F is given by the fact that, by definition, $\mathbf{S}$ remains a level set of F over time. Therefore,

$$\frac{\partial F(\mathbf{S})}{\partial t} = -\nabla F(\mathbf{S}) \cdot \frac{\partial \mathbf{S}}{\partial t} = -\left|\nabla F(\mathbf{S})\right| \frac{\partial \mathbf{S}}{\partial t} \cdot \mathbf{N} \qquad (4.31)$$

where $\mathbf{N} = \frac{\nabla F}{|\nabla F|}$.

By substituting Equation (4.30) into Equation (4.31) and re-expressing all the terms in terms of F, the evolution of explicit 3D deformable models is turned in implicit form

$$\frac{\partial F(\mathbf{x})}{\partial t} = w_1 G(\mathbf{x})|\nabla F| - 2w_2 H(\mathbf{x})|\nabla F| - w_3 \nabla P(\mathbf{x}) \cdot \nabla F \qquad (4.32)$$

in which $G(\mathbf{x})$ is inflation speed, $H(\mathbf{x}) = \nabla \cdot \frac{\nabla F}{|\nabla F|}$ is level set curvature and $P(\mathbf{x})$ a scalar potential whose gradient is the external force. The weights w_1, w_2 and w_3 are user-defined parameters controlling the influence of the respective terms. The evolution equation for implicit deformable models is solved over the whole image domain by means of upwind finite differences, a numerical approximation scheme which takes into account the direction of propagation of level sets when computing derivatives, avoiding smearing out of the solution. Once an implicit deformable model has completed its evolution, contouring can be performed on $F(\mathbf{x})$ at its zero level to obtain an explicit representation of the vessel surface. An example of level set evolution is shown in Fig. 4.11.

Implicit deformable models offer some advantages relative to their explicit counterparts. The most evident among these is topology independence, as there is no restriction on the evolution of the model in terms of representable topology or topology changes: a branch can merge with a neighbour at no computational cost or increased algorithmic complication during evolution. Another key advantage is that large deformations can be achieved without the need of re-parameterisation, since deformation is simply expressed as a change in the value of $F(\mathbf{x})$. Moreover, the evolution equation does not depend on the dimensionality of the problem, so an implicit deformable model can easily be described in N dimensions (e.g. space and time), allowing to write evolution equations for segmenting 3D time-resolved datasets. One last advantage is that the notion of surface interior and exterior is known throughout the evolution, making it easier to eventually incorporate region-based terms in the evolution equations.

Computational cost is one of the disadvantages of this approach, since in the general level sets framework a whole 3D volume must be evolved to

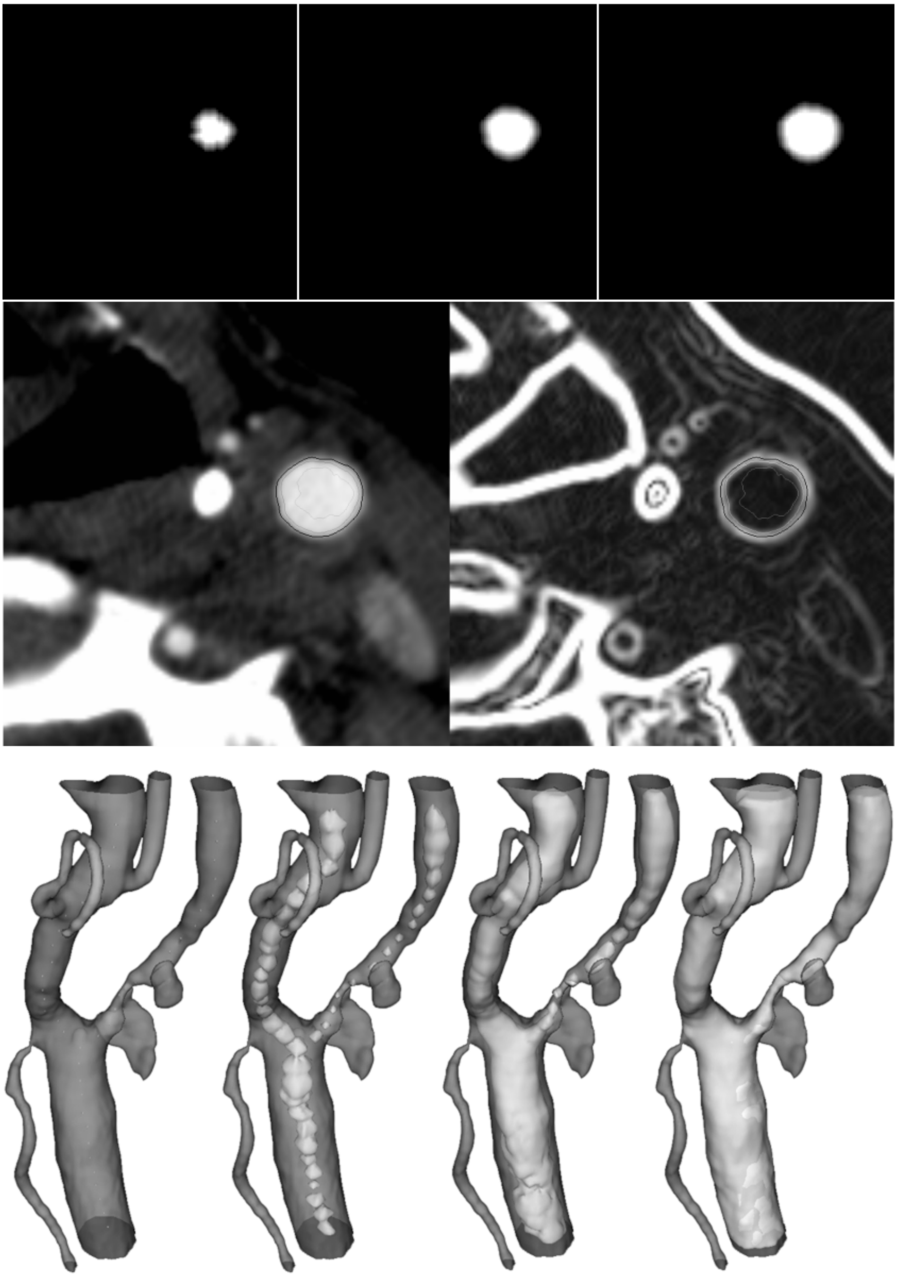

Fig. 4.11. Evolution of level sets under inflation dependent on image gradient magnitude. The top row shows the evolution of the embedding function $F(\mathbf{x})$; the mid row the corresponding zero level sets over the image (top left) and the image gradient magnitude (top right); the bottom row shows the 3D results of level set evolution, initialised from a set of disjoint seeds

track a 2D surface. In order to gain in efficiency, sparse field and narrow band methods have been proposed [540], in which only a few layers of grid points are evolved around the level set of interest, making computational cost depend on the 2D surface size rather on the size of the whole 3D embedding. One more disadvantage, which is a flip side of topology independence, is the fact that control over topology (e.g. in order to avoid self intersections or merging of neighbouring vessels) is not handled naturally by the framework.

Driving forces for deformable models

Here we now describe the image-based external force terms driving the evolution of both explicit and implicit deformable models by inflation and attraction. As a general modelling strategy, inflation terms serve to make the deformable model approach the vessel boundary from its initial shape, while attraction terms are used to make the model converge on the vessel boundary. Inflation speed terms must be crafted so that the expansion is faster in homogeneous regions (i.e. inside the lumen) and slower on vessel boundaries, while attraction terms have to be expressed as vectors pointing to image features with sub-voxel precision. In this framework, it must be noted that the role of inflation terms can be partially or completely replaced with proper initialisation strategies, such as thresholding, region growing or front propagation.

The formulation of external forces depends on the definition of image feature, which in our case must coincide with the boundary of the vessels. The most widely employed image feature definition for the segmentation of angiographic images is image gradient magnitude, $|\nabla I(\mathbf{x})|$, which quantifies the variations in local image intensity and thus highlights tissue interfaces, such as vessel boundaries. Being based on first derivatives of image intensity, gradient magnitude is likely to be sensitive to noise. Moreover, the *capture radius* of gradient magnitude can be small, that is, the model surface must be close to the vessel boundary for a gradient magnitude-based force to have an effect. For these reasons, smoothed versions of gradient magnitude are eventually employed, generated by convolving $I(\mathbf{x})$ with the gradient of a Gaussian kernel with width σ. This reduces noise and increases the capture radius to σ. Alternatively, in order to reduce the effect of noise, edge preserving smoothing can be employed [309]. In any case, the use of effective initialisation strategies can mitigate the need for an increase in capture radius and decrease the influence of noise in the deformable model evolution.

For the formulation of inflation speed terms, it is necessary to associate lower speeds to higher gradient magnitude values, as shown in Fig. 4.12. Possible choices are therefore

$$G(\mathbf{x}) = \frac{1}{1 + |\nabla I(\mathbf{x})|} \tag{4.33}$$

or

$$G(\mathbf{x}) = e^{-|\nabla I(\mathbf{x})|}. \tag{4.34}$$

As alternative formulations of the speed term, we cite the use of the function $V(\mathbf{x})$ previously presented in Section 4.3.3. With this approach, the deformable model is allowed to inflate at a higher speed in regions of high $V(\mathbf{x})$, while its inflation is slowed down as V decreases. Model speed is therefore adequately expressed by $G(\mathbf{x}) = V(\mathbf{x})$, as in [119]. In [519], the authors employ an intensity-based speed term by first fitting two normal distributions, $\mathcal{N}_v$ and $\mathcal{N}_b$, to the histograms of CE-MRA images, modelling vessel and background voxel statistics separately. Inflation speed is then expressed by

$$G(\mathbf{x}) = \frac{p_v(\mathbf{x}) - p_b(\mathbf{x})}{p_v(\mathbf{x}) + p_b(\mathbf{x})} \tag{4.35}$$

where p_v and p_b are the probabilities of a voxel of belonging to background and voxel classes, respectively, computed on the basis of $\mathcal{N}_v$ and $\mathcal{N}_b$. In another work, van Bemmel et al [520] translated the FWHM approach used in clinical settings (see Section 4.4.2) into a level set term which replaces the usual gradient based inflation speed term. The justification is to increase robustness to noise and outliers, although constraints on image acquisition modality (CE-MRA) and on the absence of flow artifacts have to be introduced. The proposed speed term is

$$G_{\mathrm{FWHM}}(\mathbf{x}) = -1 + \frac{2}{\sigma\sqrt{2\pi}} \int_0^{I(\mathbf{x})} e^{-\frac{1}{2}(\xi - \frac{1}{2}\frac{I_{\max}(\mathbf{x})}{\sigma})}\, d\xi \tag{4.36}$$

that is the error function scaled between -1 and 1. $I_{\max}(\mathbf{x})$ is the maximum intensity value in a circular region measured in a plane perpendicular to the vessel centreline.

In contrast to inflation speed terms, attraction terms are usually formulated as gradients of a potential. Model evolution is aimed at minimising energy, therefore at seeking potential minima. Such potential can be expressed as a decreasing function of image features. The classical formulation is made in terms of image gradient magnitude as $P(\mathbf{x}) = |\nabla I(\mathbf{x})|$. Regions corresponding to at greater spatial variations in intensity levels are therefore valleys of the potential. The resulting image-based force field

$$\nabla P(\mathbf{x}) = \nabla |\nabla I(\mathbf{x})| \tag{4.37}$$

has its vectors pointing towards the valleys of $-|\nabla I(\mathbf{x})|$ (as shown in Fig. 4.12). This formulation therefore translates the criterion according to which the transition between two tissues coincides with the steepest variation in intensity values. Although this is not necessarily verified for all acquisition modalities and resolutions, this constitutes a well-accepted and rather accurate criterion. Again, the distance at which the force field acts depends on the width of the ridges, which can be increased by smoothing, at the price of a lower accuracy in the identification of boundaries. A multiresolution approach can also be employed in which the width σ of smoothing kernels is initially high, and is then decreased to regain accuracy.

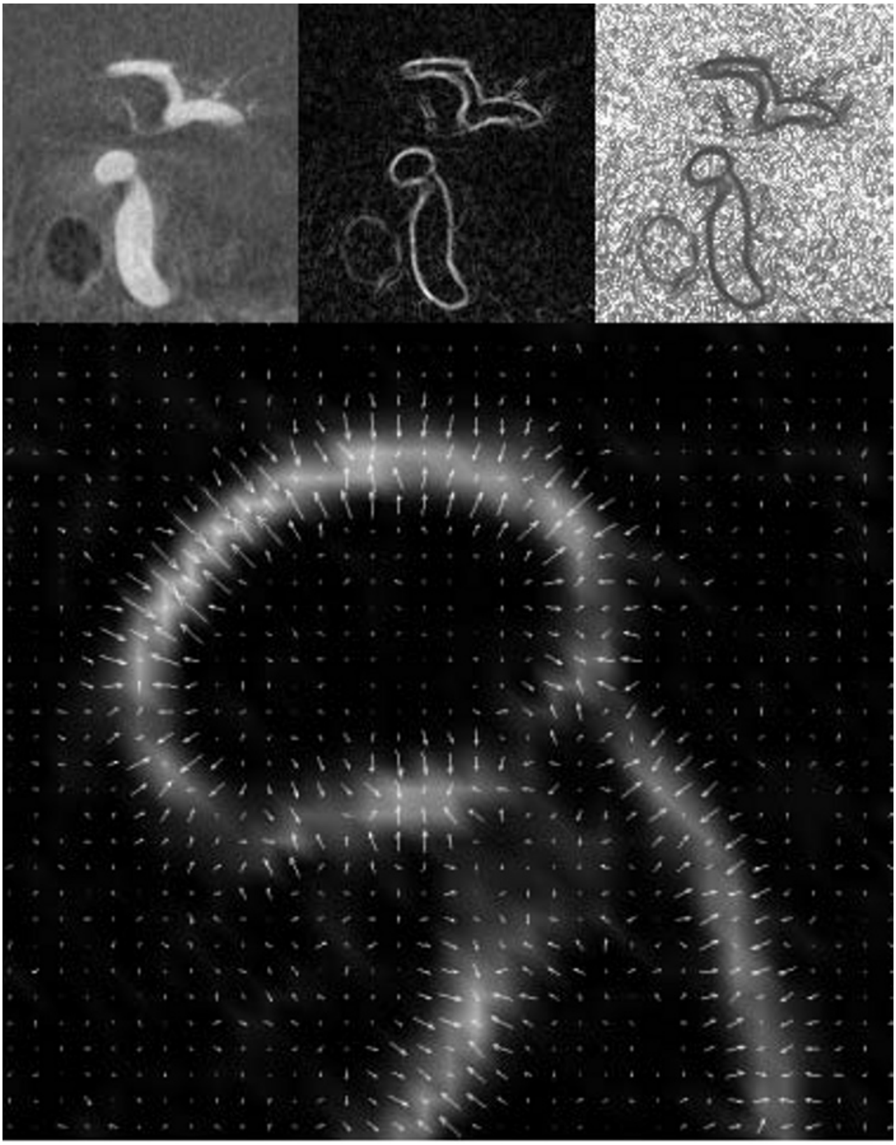

Fig. 4.12. Top left, original image; top mid, gradient magnitude image; top right, inflation speed image, obtained as $(1 + |\nabla I(\mathbf{x})|)^{-1}$; bottom, attraction field vectors $(\nabla(|\nabla I(\mathbf{x})|))$ superimposed to the gradient magnitude image, pointing to the ridges of gradient magnitude

4.5 Surface reconstruction

As extensively showed in the previous sections, the identification of the geometry of vascular structures on images can be either performed directly in 3D, or performed on a sequence of 2D images and subsequently reconstructed in 3D. In this latter case, the position of the vascular surface is known at disjoint locations, such as at a series of lines or a set of points in space, e.g. arising from tracing contours or from placing points on 2D images. In these cases, continuous representations of 3D surfaces must be generated from sparse data, a problem commonly referred to as surface reconstruction.

4.5.1 Lofting

Lofting is the procedure by which an explicit representation of a continuous surface is obtained from a set of disconnected lines. Such lines can for example be obtained as contours of the vascular lumen identified on different 2D imaging planes. These can in turn be parallel to each other, such as in conventional MR or CT, or non-parallel, as in intravascular ultrasound or freehand 3D ultrasound.

A simple lofting procedure consists in connecting each couple of successive contours by straight lines defining triangle edges. Corresponding triangle vertexes can be located along the contours based on their mutual distance, although this criterion can lead to distortion of the resulting surface in presence of abrupt changes in the shape of successive contours. One more difficulty is the generation of the lofted surface in presence of bifurcations, when one contour in one plane splits into two in the following plane.

A similar lofting technique consists in fitting a bivariate polynomial surface patch to the contour set. The surface patch is defined as a tensor-product bivariate polynomial $\mathbf{S}(u,v)$ where the parametric space is rectilinear in the longitudinal direction, $u \in [0,1]$, and periodic in the circumferential direction, $v \in [0,2\pi]$. The patch is fitted along the whole vessel with a least squares technique. Similar to the lofting procedure described above, geometric criteria are introduced in order to avoid that the circumferential parameterisation twists around the vessel [536]. The smoothness of the polynomial function allow the contours to be spaced by a greater amount than in the linear lofting case described above. This technique can notably deal with contours lying on non parallel planes, such as those obtained from IVUS images (see Section 4.1).

As already noted, the lofting approaches introduced so far are not suited to handle bifurcations. In other words, the topology of the surface has the requirement of being cylindrical, that is, it must be parameterisable by a rectangular parametric space in which one coordinate is periodic. Several approaches have been introduced to deal with the problem of bifurcating contours.

A technique used in [536] consists in fitting each bifurcation branch separately and prolonging it into the parent vessel. The interpenetrating surfaces are then clipped and stitched at the intersection lines. This method has the disadvantage of yielding non-smooth surfaces at seam lines. For this reason, this method is not recommended for modelling Y-shaped bifurcations, while it can be used to model T-shaped branches in which the parent vessel is much larger than the side branch. Even in this case, the non-realistic discontinuities in the orientation of normals at seam lines lead to models in which locations of particular physio-pathological interest, such as branches, are non-realistically reproduced.

Generic lofting algorithms capable of handling complex topologies are available in several CAD packages, and permit the user to generate free-form surfaces by controlling the shape of a set of contours. As such, they can be employed to generate surfaces from contours delineated on medical images. However, the generic nature of these solutions requires a certain degree of interaction and user skill to generate the desired surfaces, such as the placement of landmarks and editing of surface topology, at the expense of operator-independence and reproducibility.

A lofting technique that has gained popularity for 3D reconstruction of anatomical shapes from sets of parallel contours is the one proposed by Geiger [185] and implemented in the public domain software Nuages. Contours are first filled with triangles by means of a Delaunay tessellation approach (see next paragraph and Section 4.11) [165]. Subsequently, triangles on every contour are connected to the adjacent contour to form a set tetrahedra whose boundary, after a clean-up step, forms the final surface. Since one layer of tetrahedra is generated between each couple of contours, the smoothness of the final surface depends on the spacing between contours. This aspect can be improved with the use of subdivision methods in a post-processing step.

One more possibility for obtaining a continuous surface from a series of parallel contours is represented by *shape based interpolation* [422]. This method generates an implicit surface representation either from a series of closed contours or from a series of binary images generated using a segmentation technique. The implicit representation is given as a set of 2D images containing the value of the minimum distance of each pixel to the contours, referred to as the distance function. Since the contours are assumed to be closed, each distance value is given a positive or negative sign depending on whether the pixel is located inside or outside the contour, resulting in the so-called signed distance function. The computed signed distance function is then interpolated in 3D space by means of trilinear or higher-order interpolants. The final surface is the zero level-set of the resulting interpolated signed distance field.

4.5.2 Polygonisations of point sets

In a typical surface reconstruction, the position of the surface can be known at isolated points whose topological relationships are not available. As an example, this occurs when the geometry of a physical replica of a vascular segment is acquired by laser scanning techniques, but it can also occur in image-based modelling in those cases in which it does not make sense to define 2D contours due to complexity of the underlying shape (e.g. in case of a vascular network whose segments are diversely oriented with respect to the imaging plane). In this case, a continuous representation of the surface has to be reconstructed from the cloud of unorganised points.

A solution to this problem has been proposed in [226], and consists of constructing the signed distance function D to the surface approximated by the data points and then extracting the reconstructed surface by contouring D. In order to compute D, a set of tangent planes is first constructed at each data point by least squares fitting of the positions of the neighbouring points. The normals to the tangent planes are then consistently reoriented with a graph optimisation approach. The signed distance function is then obtained as

$$D(\mathbf{x}) = (\mathbf{x} - \mathbf{o}_i) \cdot \mathbf{n}_i \tag{4.38}$$

where $\mathbf{o}_i$ and $\mathbf{n}_i$ are the origin and normal of the i-th plane which is closest to the evaluation point $\mathbf{x}$.

An alternative solution for surface reconstruction from scattered points is offered by tessellation methods, which consist in filling space with simple 3D elemental volumes, typically tetrahedra, whose vertexes are the data points. The tessellation is performed in such a way that the boundary of the set of tetrahedra approximates the surface the data points have been obtained from.

At the basis of most tessellation methods is Delaunay tessellation. A tetrahedralisation of a set of points is a Delaunay tessellation when no point falls inside the circumsphere of any tetrahedron in the set. Delaunay tessellation and its dual geometric construction, the Voronoi diagram, share a variety of interesting geometric properties and they find wide application in reconstruction and shape description, see for example [165] where several methods for constructing the Delaunay tessellation of a set of points in 3D, are also described. By construction, the Delaunay tessellation of a set of points results in a convex set of tetrahedra, called a *convex hull*, irrespective of the original shape of the object whose surface points have been sampled. Therefore, its boundary does not in general correspond to the boundary of the object. What makes Delaunay tessellations attractive for surface reconstruction is the fact that, if the sampling density is sufficiently high, the original surface is approximated by subset of the faces of Delaunay tetrahedra. This means that it is possible to obtain the reconstructed surface by properly peeling external tetrahedra off the convex hull. The challenge here is the correct identification of external tetrahedra, since such information must be inferred without the help of topological information. In case the orientation of normals to the original surface is known for every point, a tetrahedron T can be classified as external when

$$(\mathbf{p}_j - \mathbf{c}) \cdot \mathbf{n}_j \geq 0 \qquad \forall \mathbf{p}_j \in T \tag{4.39}$$

where $\mathbf{p}_j$ is the j-th vertex of T, $\mathbf{c}$ its circumsphere and $\mathbf{n}_j$ the known normal to the j-th vertex. After elimination of external tetrahedra, the boundary of the resulting set of tetrahedra constitutes the reconstructed surface. Conversely, in case surface normals are not known the reconstruction method has select external tetrahedra by inferring the object shape from the 3D geometric distribution of data points. A vast literature deals with this problem (see for example [8, 9, 125, 131]).

Last, we mention the method of Boissonnat and Cazals [44], which follows the same idea of Hoppe's method described at the beginning of this section [226], but utilises the Delaunay tessellation of data points to infer neighbourhood relationships and robustly construct the signed distance function.

4.5.3 Interpolation by radial basis functions

It has already been described in 4.2.2 how a 3D surface can be implicitly represented as a superposition of analytical radial basis functions. Each RBF is centred at a predetermined location in space, which is in general associated with the known location of the surface to reconstruct, and has an expression chosen among the possibilities listed in Equation 4.11. Once the location and the expression of RBFs are determined, the implicit surface is fully determined by the coefficients w_i in Equation 4.10, as well as by the low-order polynomial function $P(\mathbf{x})$ (As suggested in [375], the latter can be omitted if the number of data points is large ($N > 10$); for this reason, in the following we will assume $P(\mathbf{x}) = 0$).

At this point, the problem is that of determining coefficients w_i from a set of N data points $\mathbf{p}_i$ of known location such that the resulting embedding function F is zero at the data points and smoothly interpolates between them. This implies the imposition of a set of conditions of the type

$$F(\mathbf{p}_i) = d_i \qquad i = 1, \ldots, N \tag{4.40}$$

which leads to a linear system of equations of the form

$$A \cdot \mathbf{w} = \mathbf{d} \tag{4.41}$$

where $A_{ij} = \phi(|\mathbf{p}_i - \mathbf{p}_j|)$ is a symmetric $N \times N$ matrix, $\mathbf{w}$ is the vector of N weights w_i and $\mathbf{d}$ is the right hand side vector containing the values that F has to take at the data points. The matrix A is non singular if $N \geq 2$ and $\mathbf{p}_i \neq \mathbf{p}_j$ for $i \neq j$. In theory, since the data points should lie on the surface, $\mathbf{d}$ should be identically zero. In practise, this would lead to a linear system that, if non singular, would only admit the trivial solution $\mathbf{w} = 0$. Therefore, one or more data points must be specified in such a way that $d_i \neq 0$. These points are referred to as *offset points*, and can be generated by moving data points away from the surface of a fixed amount. As an example, if data points are generated from closed planar contours, offset points can be generated by shrinking the contours along their normals of a fixed amount, as shown in [375].

Depending on the choice of RBF and the distribution of data points, the system needed for the computation of $\mathbf{w}$ can be badly conditioned or not positive definite. For this reason, robust iterative methods such as GMRES are employed. In order to improve convergence properties, ad-hoc preconditioners or regularisation techniques have been proposed [375].

In general, the choice of linear or cubic RBFs (see Equation 4.11), leads to smooth implicit functions and good convergence rates.

4.6 Overview of vascular modelling strategies

In the previous sections we briefly overviewed some of the main techniques available for accomplishing the several steps involved in modelling vascular segments from medical images. We will now give an overview on how these techniques have been combined to provide full-fledged image-based modelling strategies.

As anticipated previously, the choice of a modelling strategy depends on image modality (2D vs. 3D, bright blood vs. black blood), image quality (presence of noise or artifacts), vascular segment geometry and topology (simple vs. complex), purpose of analysis (pure geometry, hemodynamics or fluid-structure interaction), and it must take into account the issues of accuracy, reproducibility, robustness, automation. Automation has in general been considered a key feature in proposing tools for clinical contexts, such as for stenosis grading, while accuracy has been assigned a higher priority over automation for applications such as geometric or haemodynamic modelling, at the price of user interaction. In some situations the intervention of a trained operator is hardly replaceable, like in the case black blood MR imaging, when the presence of complex haemodynamic patterns gives rise to plaque-mimicking flow artifacts which are difficult to automatically discern from the vessel wall.

As already mentioned, the dimensionality of the imaging modality contributes to the choice of the dimensionality of the technique. For example, if the acquired images are made up of strongly anisotropic voxels, for which in-plane resolution is much greater than along the imaging plane normal, 2D segmentation followed by a 3D reconstruction offers more flexibility in handling typological and geometric changes occurring from one slice to the next. On the other hand, 3D segmentation provides a more straightforward possibility on datasets with a more isotropic resolution, allowing modelling of vascular segments with complex geometry. In general, initialisation and specification of user-defined constraints require more sophisticate approaches if directly performed in 3D.

Two representative studies in which vascular models were generated by 2D segmentation followed by 3D reconstruction are Milner et al [333] and Long et al [297]. In the former, carotid bifurcation models were generated from black blood MR images. 2D outlines were first generated by means of a region growing algorithm, which provided a sequence of oriented points around the lumen. Point position was eventually interactively adjusted based on the degree of fit to the wall and smoothness of a B-spline surface fitted to the points. Contours were finally lofted with B-spline surface patches, one for each vessel branch (common, internal and external carotid artery). Each surface patch was bounded by a contour; in particular, all surface patches were attached to the same contour at the bifurcation level, so to ensure surface continuity. Long et al instead resorted to 2D snakes for the segmentation of TOF-MRA images or abdominal aorta bifurcations. A two-stage smoothing procedure was applied to the generated lumen contours in order to correct misalignment of subsequent contours due to subject movement, and to improve global surface appearance. Four surface B-splines were generated from lumen contours, two of them following the outer wall of the bifurcation, from the parent vessels to the two daughters respectively, and two on the inner side of the daughter branches, meeting at the bifurcation apex.

A 2D segmentation technique geared towards the delineation of both lumen and wall boundaries in black blood MR images was presented by Ladak et al [266], based on a DDC model. Steinman et al [482] employed this method to generate 2D contours of carotid bifurcations. The contours were then filled to create a binary volume, inside which a 3D balloon was inflated, in order to generate a 3D mesh of triangular elements, whose smoothness was imposed through internal forces. In the same work, owing to the acquisition technique employed, the outer boundary of the vascular wall was also segmented using the same deformable model, and a 3D model of the vascular wall was derived. Besides allowing to test hypotheses on the correlations between hemodynamics and wall thickening, as introduced in Chapter 1, this technique might in principle be used for generating fluid-structure interaction models (see Chapter 3 and 9) based on patient-specific lumen and wall geometry.

Wang et al [536] addressed the problem of performing 2D outline generation for geometrically complex vessels by extracting approximated vessel centrelines from 3D image datasets, generating images oriented normally to vessel centrelines and performing level set-based 2D segmentation on the resulting images. The outlines were then lofted to produce analytical explicit representations of single branch surfaces, which were then merged together to obtain a complete model of a branching vascular structure.

Giordana et al [194] employed a 2D segmentation method based on edge detection and thresholding followed by cubic spline fitting to segment TOF-MRA images of distal bypass grafts. 3D reconstruction was then performed by means of implicit RBF. The same group [375] recently proposed automated segmentation of TOF-MRA by means of Otsu's thresholding followed by RBF reconstruction.

Kaazempur-Mofrad et al [248] employed nonlinear image enhancement and snake-based 2D segmentation followed by 3D surface lofting to reconstruct diseased carotid bifurcations from black blood MR images.

As for 3D explicit deformable models, Ladak et al [265] proposed the use of balloons initialised as a simple 3D shape (e.g. a sphere) and then evolved with a two-stage procedure, during which the model is first inflated under the effect of a gradient-dependent term until it approaches the vessel wall and then optimised under the effect of a gradient-based advection field. This technique was successfully employed for the segmentation of a cerebral aneurysms from RA [478] which constituted the first patient-specific model of hemodynamics in cerebral aneurysms. As an alternative to this approach, Yim et al [550] proposed to initialise explicit deformable models with isointensity surfaces built using a 3D contouring algorithm directly on the images. This provides easier generation of topologically complex initialisations, although the exclusion of unwanted structures requires further processing.

Last, 3D implicit deformable models have been employed in [119] [12, 518] with different initialisation strategies. In [119], a front propagation approach based on the solution of the eikonal equation was employed in order to initialise level sets near the vessel wall. In [518] initialisation is provided by the estimation of centrelines performed on the basis of vesselness (see Section 4.3.3), and image-dependent inflation terms are then employed in conjunction with stopping criteria. In [12], centrelines were estimated from a surface extracted by contouring and again employed for initialisation of level sets. A first inflation stage was then performed, followed by attraction of level sets to vessel boundaries. Segmentation was carried out separately for branches of different scales, in order to avoid scaling problems in the evolution parameters. Final level sets relative to different branches were then merged together to obtain the complete model.

In short, the problem of defining the geometry of a computational domain representing a vascular segment from medical images has several possible solutions. The choice has to be made based on image characteristics and on the expected geometry and topology of the vascular segment under consideration. Since the outcome of this step is the definition of the computational domain upon which the modelling problem will be carried out, a careful choice of the technique and of the involved parameters is of primary importance for ensuring the significance of the final predictions.

4.7 Mesh Generation

The last step in the generation of computational domains from image data for the numerical approximation of PDEs introduced in Chapter 2 and 3, is the discretisation of the physical domain in small elements of simple shape and finite size (the computational mesh), as illustrated in Fig. 4.1.

To this end, this section contains a brief description of available mesh generation techniques that we consider to be particularly useful for cardiovascular geometries. The reader should be aware that there is no such thing as a “general” mesh generation technique; the most efficient or appropriate technique for mesh generation depends on the geometrical and physical complexity of the problem being solved as well as on the method of discretisation chosen. In what follows, the term mesh will be used to refer to the spatial discretisation required for the numerical approximation of PDEs. The term grid is often used in the literature with the same meaning, but here the word grid is reserved to denote a lattice of voxels/pixels of an image, as in the previous sections.

The literature in mesh generation is extensive. The reader could consult the general review articles [365,474,501], the textbooks [172,260,499,500], the special journal issues [59, 366], and the “Meshing Research Corner” website maintained by S. Owen to expand on the necessarily brief coverage of the topic provided here.

4.8 B-Rep model of the computational domain

A general framework for the definition of a computational domain is the so called boundary representation, or B-Rep, where the domain is viewed as a polyhedron bounded by curved faces defined on surfaces. The faces in turn are polygons bounded by edges defined on curves that, in what is commonly referred to as a manifold representation, are common to two surfaces (see Fig. 4.13). Non-manifold representations are discussed elsewhere, e.g. [312].

The surfaces in the B-Rep definition of the computational domain can be defined as NURBS surfaces, implicit or level set surfaces, or triangulations, as discussed previously in Section 4.2. The edges are represented by the intersections between these surfaces.

A common practise is to generate the mesh in a bottom-up fashion, namely, starting from a division of the edges into segments, followed by a discretisation of the faces, and finally the generation of a volume mesh in the interior of the domain.

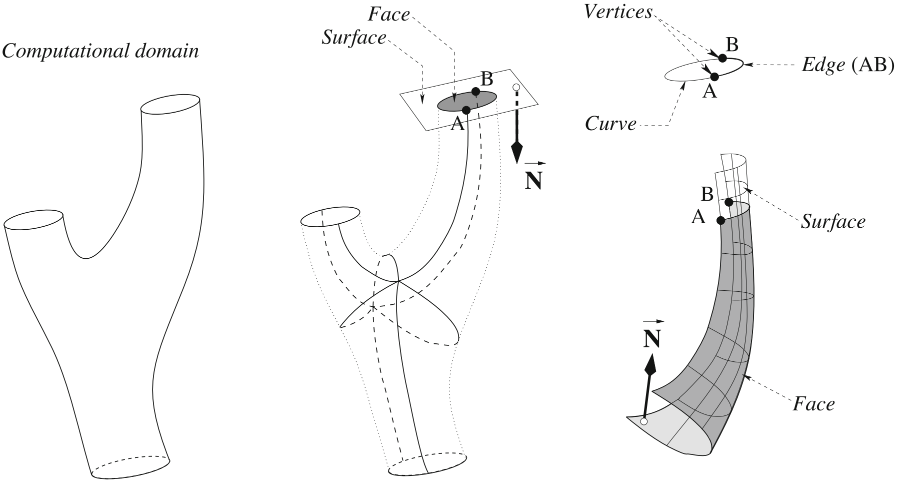

Fig. 4.13. Boundary representation (B-Rep) model of the computational domain

4.9 Structured versus unstructured meshes

Structured meshes are usually associated with numerical approximations by finite differences. The nodes of such meshes follow a regular repeatable pattern where each interior node is always connected to the same number of neighbouring nodes, as shown in Fig. 4.14(a). The presence of such a pattern means that the data structures required to represent the mesh are trivial, and the neighbour nodes can be easily identified by a suitable numbering of the nodes consistent with this pattern.

On the other hand, unstructured meshes do not show such a pattern and the number of neighbours to a node changes from node to node in the mesh. An example of such meshes is shown in Fig. 4.14(b). The data structures required to represent unstructured meshes are more involved as we need to store additional information to identify the neighbours of each node in the mesh. Finite elements and finite volumes are the techniques commonly used for numerical simulations in this type of meshes.

The mesh structure affects the efficiency of numerical algorithms for the solution of PDEs. The neighbouring values required for the construction of numerical approximations at a node of a structured mesh can be easily identified by direct addressing in the array storing the variables. A similar construction on an unstructured mesh requires the use of a connectivity array to identify the nodes neighbouring a given node. Its implementation requires indirect addressing of the array of variables. The implications for efficiency are significant: loops implemented using indirect addressing are at least twice as expensive as similar loops where direct addressing is used. Therefore, whenever the complexity of the geometry permits the generation of an structured mesh in a reasonable time, we could use numerical algorithms that take advantadge of this structure to produce solvers that are faster and more efficient than those implemented on unstructured meshes.

In what follows and without significant loss of generality, we will restrict our discussion mainly to structured meshes that are topologically square lattices of points and to unstructured meshes of triangles and tetrahedra. Other lattice topologies and element types are possible but we refer to the literature cited in Section 4.7 for a discussion.

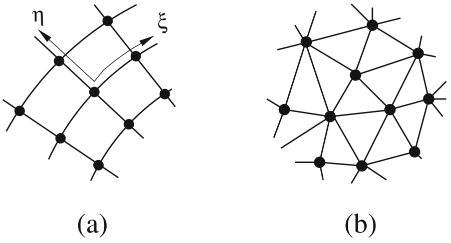

Fig. 4.14. Structured and unstructured mesh discretisation of a computational domain

4.10 Structured mesh generation

Finite difference approximations are easily implemented on a Cartesian system of reference by defining an orthogonal network of lines parallel to the axes. The mesh nodes then correspond to the intersection of the coordinate lines. The application of boundary conditions is easy if the nodes are located on the boundary. However, this is often not the case in the presence of curved boundaries. It is still possible to impose boundary conditions at the intersection points of the network of lines with the boundary. This means that the finite difference approximations use unevenly distributed points which typically increases the truncation error of the approximation at boundary when compared with that of the interior points where the intervals are evenly spaced. Neumann boundary conditions, such as those described in Chapter 2, are difficult to implement since they involve the normal to the boundary and require additional points to construct the approximation.

To reduce the errors when applying boundary conditions one could use higher order approximations at the boundary, but this increases the computational cost and complicates the implementation as requires additional bookkeeping to construct the boundary approximations. An alternative method is to increase the resolution at the boundary using *recursive subdivision* methods, called *quadtree* in 2D and *octree* in 3D for reasons that will become clear in the next section.

4.10.1 Quadtree and octree Cartesian meshes

The main idea behind these methods is to selectively refine the mesh, i.e. reduce the mesh size, near the boundary to reduce approximation errors. The procedure starts by defining a coarse Cartesian mesh (e.g. 3 × 3 points) which contains the domain Ω. Cells that contain the boundary are identified and, if the discretisation error is larger than a certain prescribed value, the cell is divided into four. This subdivision is recursively applied to the newly created cells until the discretisation error is such that no further subdivision is required, as illustrated in Fig. 4.15.

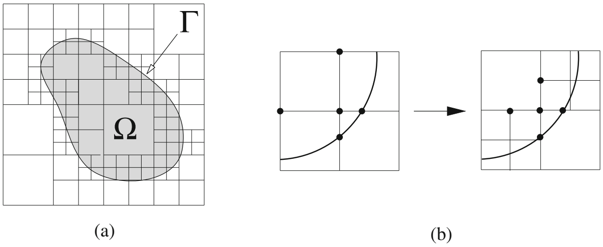

Fig. 4.15. Quadtree structured mesh: (a) general view; (b) improving the truncation error via subdivision near boundaries

An efficient way of implementing the method is the use of a tree structure in which the initial four cells are at the root and each cell is associated either the four cells that result from a splitting by half in both directions, or none if the cell is not broken up any more. These type of structures are well known and documented in the computer science literature [446], where they are extensively employed, e.g., for compiler parsing and image processing.

The use of a quadtree structure increases the amount of memory required as the tree needs to be stored. The discretisation of the PDE requires a complex housekeeping which involves a tree searching to form the finite difference approximations. The ideas described in 2D for the quadtree structure are readily applied in 3D (octree), where each cell is now subdivided into eight.

The added complexity required for the treatment of curved boundaries and the difficulty in applying boundary conditions in the normal direction has limited his application for viscous flows on complex geometries.

The favoured approach for structured meshes in the presence of curved boundaries is the use of *boundary conforming* meshes. These are meshes in which their boundary nodes are on the boundary of the computational domain thus eliminating the need for interpolation at intermediate points and reducing the associated discretisation errors.

4.10.2 Mesh generation by mapping

A method for generating structured boundary conforming meshes is the use of a suitable mapping. The idea of using a mapping $(x, y) \rightarrow (\xi, \eta)$, yet to be defined, is therefore to transform the domain Ω with boundary Γ into a rectangular region Ω^* with boundary Γ^* as shown in Fig. 4.16.

The original PDE is also transformed into a new PDE, which is solved in a square region where the application of boundary conditions is greatly simplified and the discretisation can be easily and accurately performed using standard finite difference approximations. An alternative option is to use the inverse mapping to transform a network of orthogonal lines in Ω^* into a net of coordinate lines (a mesh) in Ω which is boundary conforming. This mesh in Ω is boundary conforming and can be employed to discretise the PDE and boundary conditions using finite volume or finite element methods.

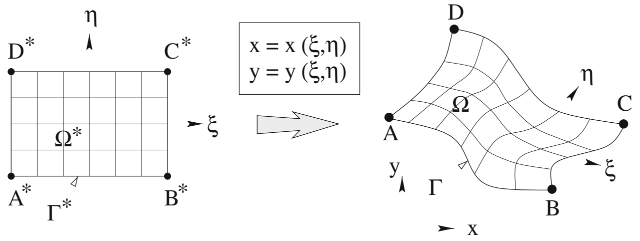

Fig. 4.16. Mapping of a region into a square region

The problem of this approach is that of finding a suitable mapping. The main types of structured mesh generation methods are either algebraic methods or based on the solution of elliptic, parabolic or hyperbolic PDEs.

Algebraic methods define the mapping via an explicit mathematical expression. For instance, the method of transfinite interpolation [134,470] obtains the mapping via the interpolation of polynomials through a set of points along the coordinate directions. Their main advantage of these methods is their simplicity and efficiency. However they often do not afford the user sufficient control to cluster points in regions where a finer mesh is required or to avoid mesh overlapping occuring for certain geometries.

Mesh generation by means of elliptic PDEs is discussed in the next section.

4.10.3 Elliptic mesh generation

The mapping that will transform a Cartesian mesh in the parameter space (η^1, η^2, η^3) into a mesh in the physical space $\mathbf{r} = (x, y, z)$ is defined as the solution of the system of elliptic PDEs of the form

$$\sum_{i=1}^{N}\sum_{j=1}^{N} g^{ij}\frac{\partial^2 \mathbf{r}}{\partial \eta^i \partial \eta^j} + \sum_{k=1}^{N} P^k \frac{\partial \mathbf{r}}{\partial \eta^k} = \mathbf{0} \tag{4.42}$$

where N is the number of dimensions and

$$g_{ij} = \frac{\partial \mathbf{r}}{\partial \eta^i} \cdot \frac{\partial \mathbf{r}}{\partial \eta^j}\,; \quad \sum_{j=1}^{n} g^{ij} g_{jk} = \begin{cases} 1 & i = k \\ 0 & i \neq k \end{cases}. \tag{4.43}$$

The terms $P^k(\eta^1, \eta^2, \eta^3)$ are mesh control functions that are suitably defined to increase or decrease the mesh resolution about nodes or lines in the mesh.

The equations 4.42 are solved numerically. The unknowns are the coordinates of the interior mesh nodes. The boundary conditions are the coordinates of the mesh nodes in the boundary obtained in the previous step of the discretisation of the B-Rep of the computational domain. The PDE is discretise using centred finite differences and the resulting system of equations solved using iterative techniques.

The method has been applied to the generation of some very complex 3D geometries. However, finding a suitable mapping or sequence of mappings to transform the topologically square computational domain into a square it is often very difficult and requires a high degree of ingenuity. A strategy that, to a large extent, remedies this problem is the use multiblock meshes.

4.10.4 Multiblock techniques

In the multiblock method the computational domain is divided into a number of subdomains or blocks that are topologically square. These subdomains are

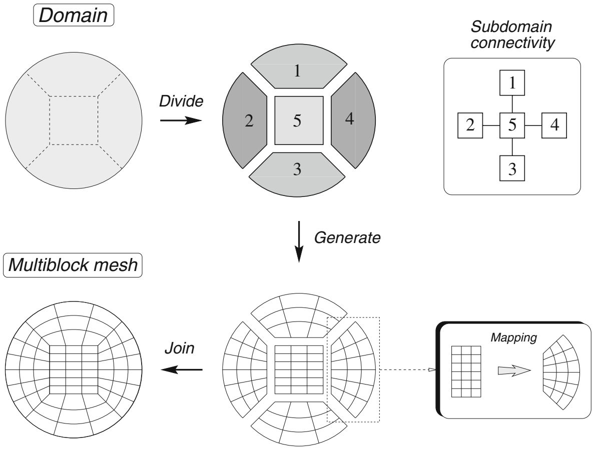

Fig. 4.17. Multiblock mesh generation

individually discretise using a mapping into a square or cube. Continuity across the interfaces between the blocks is ensured by requiring that the points (and perhaps tangent vectors) at the interfaces are shared by adjacent blocks. The different steps involved in the discretisation procedure by the multiblock approach are illustrated in Fig. 4.17.

This method has been very successful in producing structured meshes for complex 3D geometries. However, there are two main drawbacks. The first is an increase in computer memory for storage of the connectivity between the blocks and the additional information from neighbouring blocks that is required (usually in the form of halo cells) to form the approximations at the block interface boundaries. The second, and also the main bottleneck of the method at present, is the lack of a general method for the automatic subdivision of the computational domain into topologically square regions. This process often requires some form of user intervention that makes the task very difficult and time-consuming.

A way to ease the process of automatically generating meshes for complex geometries is the use of *composite overlapping* or, alternatively, *unstructured* meshes.

Composite overlapping meshes, also known as overset or chimera meshes, consist of separate boundary conforming meshes for each of the different components of the boundary, but without defining a precise interface with common points. Figure 4.18 shows a simple 2D example.

These methods facilitate considerably the generation of meshes for very complex geometries, but require the use of special interpolation procedures to transfer information between meshes. This could result in a loss of accuracy at the interfaces. For instance, properties such as conservation are difficult to preserve, particularly in 3D. Further information about these methods can be found in [81].

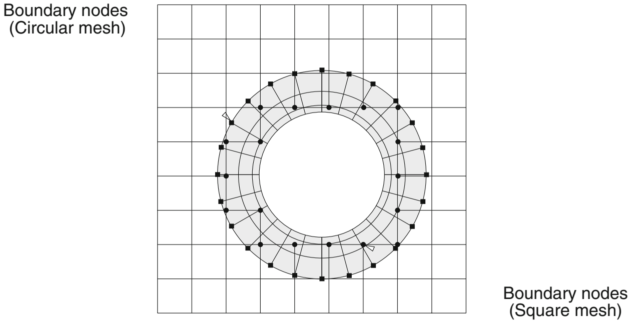

Fig. 4.18. Composite overlapping meshes

As discussed in Section 4.9, unstructured meshes do not have a predefined node and element arrangement, this lack of structure eases greatly the mesh generation and permits a better automatisation of the process. Their use has lead to a considerable reduction of the time required to generate a mesh for a complex geometry as compared to multiblock.

4.11 Unstructured mesh generation

In this section we discuss three of the most commonly used methods for generating unstructured meshes: Delaunay, advancing front and octree.

4.11.1 Delaunay mesh generation

This method is based in the construction of the Voronoi diagram for a given set of nodes. This is a spatial partition of the domain into regions, each of which is associated to a node in the set and contains the points that are closer to that node than to any other node in the set. A Delaunay triangulation, already introduced in Section 4.5, is uniquely defined by joining, with a straight line, nodes in regions sharing an edge. An example of a Voronoi partition and its dual Delaunay triangulation is shown in Fig. 4.19(a).

An important property of the Delaunay mesh is that circumcircles (circumspheres) of a triangle (tetrahedron) do not contain other nodes in the mesh. This is depicted in Fig. 4.19(b) and it will be referred to as the in-circle test in what follows. The algorithm proposed by Watson [537] uses this property to generate a mesh as follows. Assume that we have an existing Delaunay mesh where we want to introduce a new node. Those triangles in the mesh that do not satisfy the in-circle test, i.e. their circle contains the new point, are eliminated from the mesh. A new Delaunay mesh is then generated by connecting the new node to the nodes in the boundary of the (convex) hole created in the mesh in the previous step. This process is repeated for all the nodes in the set. An illustration of the algorithm is shown in Fig. 4.20.

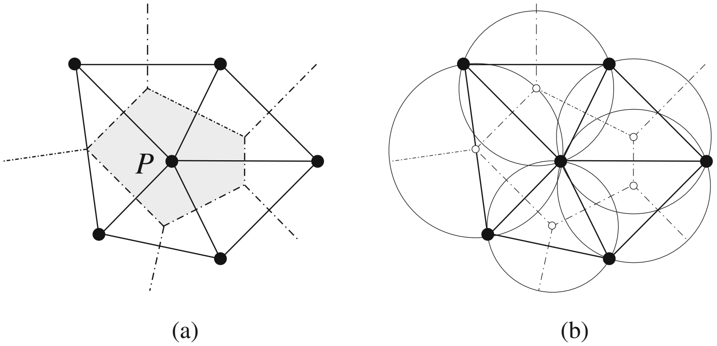

Fig. 4.19. Delaunay triangulation: (a) Voronoi diagram of a set of points. The shaded region indicates the Voronoi region associated with the point P. The dashed lines represent the boundary edges of the Voronoi regions. The Delaunay triangulation is obtained by joining those point that share a boundary edge. The Voronoi edges are the median lines of the sides of the triangulation. (b) The in-circle criterion: the circumcircles of the triangles in a Delaunay mesh do not contain any other nodes

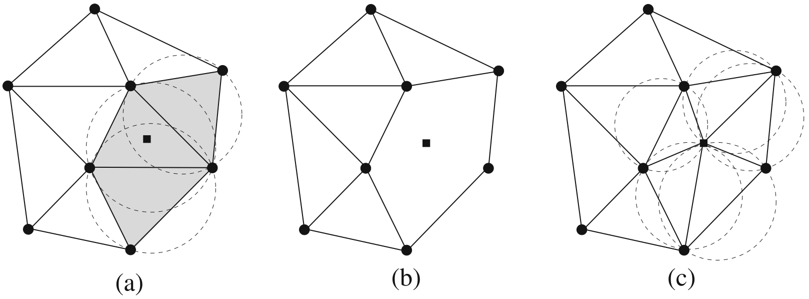

Fig. 4.20. An illustration of the Watson's algorithm for Delaunay mesh generation: (a) a new node is introduced into an existing Delaunay mesh, the triangles that do not satisfy the in-circle test are shaded; (b) the triangles that do not verify the in-circle test are deleted and (c) the new node is connected to the nodes in the boundary of the hole created to produce a new Delaunay mesh

A reliable implementation of this technique requires the use of robust algorithms that provide unambiguous answers for the in-circle test. These is required to handle degenerate cases, such as for instance four or more nodes in the same circumcircle, that will lead to topological incompatibilities (self-intersection) if the in-circle test is affected by round-off error. Robust algorithms for the in-circle test and other geometrical tests are discussed in [18, 461]. If the set of points is given, the main cost of the algorithm is the search for the circles that contain a point to be inserted in the mesh. A much more efficient version of the method is obtained is the search is reduced to a local search as proposed by [538]. The main idea is to automatically generate a point within a triangle if this is consistent with a specified mesh size distribution. As a result, the verification of the in-circle condition requires only a *local* search that is easily implemented by storing a list of neighbour triangles to a triangle.

Another very important property of the Delaunay triangulation is that its boundary is the convex hull of the set of nodes. If the boundary of the computational domain is not convex, an additional post-processing of the mesh is required for recovering a suitable discretisation of the boundary. Boundary recovery is the main difficulty associated with this generation technique, particularly if we require the mesh that conforms to the boundary discretisation to be Delaunay. This could be achieved, to some extent, by boundary point insertion but this might lead to a mesh which is too fine at the boundary. If obtaining a Delaunay mesh is not a requirement, mesh modification techniques could be employed to recover the surface discretisation [186].

4.11.2 Advancing front mesh generation

The advancing front method is based on the use of a generation front formed by facets that will be used to construct elements using a facet of the front as the element base. For a tetrahedral mesh, the facets are triangles and the elements are tetrahedra. The mesh elements are generated in sequence and, at a given stage of the generation process, the generation front is the discretise boundary of the region in the computational domain that remains to be meshed. The facets are consistently orientated so their normal could be used to determine the relative position of that region with respect to the front.

The generation front is a dynamic data structure which changes continuously during the generation process. The initial front is the set of facets generated in the previous stage of the discretisation of the B-Rep of the computational domain. During the generation process a facet is selected from the front and an element is generated. This may involve creating a new node or simply connecting to an existing node in the front. The validity of the element is verified by checking that this element does not intersect the generation front. After a valid element has been generated, the front is updated by deleting the facets of the newly generated element that are present in the current front and inserting the facets that are not. This updating process is illustrated in Fig. 4.21 for a two-dimensional domain.

The advancing front method preserves the boundary discretisation and also permits a straightforward implementation of mesh control. However, it is an heuristic method. A successful implementation requires procedures that are robust, i.e. insensitive to computer round-off error, to verify the validity of the element, a suitable strategy for the selection of facets and points to generate new elements, and the use of suitable range searching and sorting algorithms and data structures for efficiency [34].

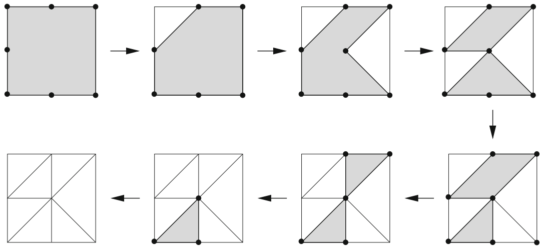

Fig. 4.21. Advancing front mesh generation. The shaded region represents the region that remains to be meshed. Its boundary is the generation front. The arrows indicate the generation sequence

The main advantadge of the method is that it preserves the boundary triangulation and thus there is no need for boundary recovery. The quality of the meshes it generates is very similar to that obtained by the Delaunay method when the same mesh distribution is prescribed. The cost is also comparable if a range search is required to insert points in the Delaunay mesh. However, the Delaunay method with automatic point insertion is more efficient.

This method can also be employed for the generation of unstructured meshes of quadrilaterals in reference [40, 559] and of hexahedra as described in references [41].

4.11.3 Octree and quadtree unstructured mesh generation

The starting point is the definition of a box, or cell, that contains the computational domain. This cell is recursively subdivided into smaller cells, usually four in 2D or eight in 3D. The subdivision is applied until the smallest cells permit to represent the boundary accurately. To ensure good mesh quality, the size of neighbouring cells is required not to differ by more than a factor of two. If the boundary discretisation is an input, then the vertices of the tree are moved to recover it. Otherwise, the intersections of the cells with the geometry are calculated and the discretisation of the boundary obtained in a manner similar to the marching cubes technique described in Section 4.2.3. In the final step, we triangulate the portions of the cells that are interior to the domain and exclude those that are exterior. An illustration of quadtree mesh generation is depicted in Fig. 4.22.

The octree technique is fast, conceptually simple and works reasonably well even with an inaccurate B-Rep of the boundaries. On the other hand, it introduces artificially preferred directions and produces poor elements near the boundary. This is due to the limited control on refinement near boundaries which must occur by factors of two. The quality of the mesh can be improved by the use of mesh enhancement techniques which are described in the next section.

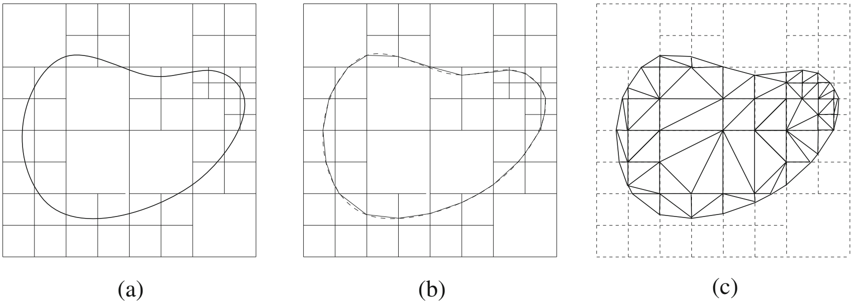

Fig. 4.22. Quadtree mesh generation: (a) spatial partition of the cells; (b) boundary discretisation; (c) triangulation of the interior portions of the cells

4.11.4 Mesh control, enhancement and adaption

The geometrical properties of the mesh play an important role in the efficiency and accuracy of numerical simulations performed on it. The cost of a flow simulation on a given mesh is proportional to the number of elements or nodes in the mesh. The quality of the flow solution at a given location is affected by the local discretisation errors that depend both on the mesh size and the smoothness of the solution. Finally, mesh distortion, e.g. the presence of very small angles, could have a detrimental impact on the convergence of algorithms for the solution of the discretise PDEs. It is therefore very important to incorporate mesh *control* strategies in mesh generation to achieve an optimal mesh, i.e. the one that produces the best accuracy for a given number of nodes.

Mesh control is usually implemented by requesting that the mesh to be generated conforms to a given spatial distribution of element size and shape. This distribution could be given as a user-defined function. Alternatively we could use techniques for mesh *adaption* where the mesh control function is automatically inferred from the geometry of the domain, a flow simulation on the current mesh, or both.

Mesh control is easily incorporated into any of the unstructured mesh generation techniques previously described. In the octree mesh generation this is achieved by splitting a cell only if the children cells have a length larger than the mesh size specified by the mesh control function at the cell centre. In the advancing front method, the characteristics of the element to be generated could be obtained by the evaluation of the mesh control function at the centroid of the front facet used as its base. In the Delaunay triangulation, a new point is inserted in the mesh only if its insertion leads to elements of a size larger than that specified by the mesh control function.

Due to algorithmic and geometrical constraints it is not always possible to achieve, using the previously described techniques, a mesh without distorted elements. The quality of an element of the mesh is almost always evaluated by comparison with its equilateral equivalent. Several quality indexes or metrics have been proposed in the literature and most of them are discussed in [462]. If a mesh control function is specified, the quality index must be computed in a suitable normalised space as described in [172, 374].

The quality of the mesh can be improved through the use of mesh *enhancement* techniques such as mesh smoothing, side splitting, side collapsing and side swapping. These techniques are of a local nature and modify only a few nodes and elements in the mesh. The enhancement of the mesh is often achieved by locally applying them in sequence whilst sweeping through the mesh.

Mesh smoothing moves a node of the mesh to a new position that is a weighted average of the positions of the nodes that belong to the elements in the mesh surrounding the node. *Side splitting* inserts a new node in a side of the mesh and divide all the elements sharing the side in two to produce elements that contain the new node. *Side collapsing* takes a side of the mesh and eliminates it from the mesh by collapsing its two nodes into a new node. As a result, all the elements sharing the side also disappear from the mesh. *Side swapping* a side is eliminated from the mesh, leaving a cavity bounded by a polyhedron formed by the faces of the elements surrounding the side. A new mesh topology for this cavity is obtained by first constructing a triangulation using the nodes in the polyhedron that did not belong to the deleted side and then connecting these triangles to the nodes of the eliminated side. There might be many ways of constructing this triangulation and the obvious choice is to select the one leading to the best mesh quality. All these procedures could lead to the generation of invalid elements for certain meshes. In practise this is avoided by applying these operations only if the modified mesh is valid.

4.11.5 High-order mesh generation

High-order elements, despite their increased computational cost with respect to linear elements, can efficiently achieve high accuracy for a given computational work. When used in the frame of a spectral element method to solve the differential problem at hand, they exhibit exponential convergence for smooth solutions as the order of the polynomial is increased, and are particularly efficient for the long time integration of unsteady flows, see for instance [60, 253].

The generation of a mesh of boundary conforming high-order elements can be achieved through the modification of a previously generated mesh by creating the additional points required for the high order interpolation and projecting them on the edges and surfaces of the boundary of the computational domain.

The construction of such elements starts with a discretisation of the edges of the tetrahedral mesh into N points as required by a Lagrangian polynomial interpolation of degree $N-1$. There are three possible cases to consider. If the end points of the mesh edge belong to an edge of the of the domain, the interpolation points are generated on the edge. If the mesh edge is on the boundary surface, the interpolation points are generated in a suitable projec-

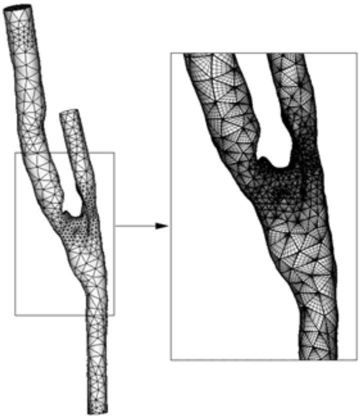

Fig. 4.23. Unstructured mesh of high-order elements: the figure on the left shows the coarse mesh representing the projection of the edges of the linear mesh on the boundary. The enlarged figure includes the interior points used in defining the high-order approximation within the elements

tion of the mesh edge onto the surface. In all other cases, the edge is taken to be straight and the interpolation points are equally spaced along its length.

The next step requires the subdivision of the faces of the tetrahedral elements. This involves two possible situations. If the mesh face is on the boundary surface, a transfinite interpolation from the interpolation points on the edges of the face followed by a projection on the surface is used to calculate their 3D coordinates. For all the other faces, a transfinite interpolation between the 3D coordinates of the interpolation points on the edges is employed to compute the position of the interior points. The final splitting of the tetrahedral elements involves a transfinite interpolation between the coordinates of the points previously generated on the faces of the element. An example of a high-order mesh is shown in Fig. 4.23. More details about these techniques can be found in [124, 459].

4.12 Conclusions

The purpose of this chapter was to give an overview on the existing techniques for generating patient-specific computational domains representing vascular segments starting from image data. Rather than providing a comprehensive list all the available approaches, we tried to organise this wide matter into a unified framework, focussing on the fundamental concepts and terminology. Our aim was to provide the reader with the means for choosing the best techniques for the specific problem at hand. What has been described in this chapter enables the application of the mathematical techniques presented in the reminder of the book to patient-specific cases.