Local implicit modeling of blood vessels for interactive simulation

Ahmed Yureidini, Erwan Kerrien, Jérémie Dequidt, Christian Duriez, Stéphane Cotin

Local implicit modeling of blood vessels for interactive simulation

A. Yureidini[1 2 4], E. Kerrien[1 3], J. Dequidt[2 4], C. Duriez[2 4], S. Cotin[2 4]

[1] Inria, Villers-lès-Nancy, F-54600, France
[2] Inria, Villeneuve d'Ascq, F-59650, France
[3] Université de Lorraine, Loria, UMR7503, Vandœuvre-lès-Nancy, F-54600, France
[4] Université Lille 1, Lifl, UMR8022,Villeneuve d'Ascq, F-59650, France

Abstract. *In the context of computer-based simulation, contact management requires an accurate, smooth, but still efficient surface model for the blood vessels. A new implicit model is proposed, consisting of a tree of local implicit surfaces generated by skeletons (*blobby models*). The surface is reconstructed from data points by minimizing an energy, alternating with an original blob selection and subdivision scheme. The reconstructed models are very efficient for simulation and were shown to provide a sub-voxel approximation of the vessel surface on 5 patient data.*

1 Introduction

In the context of interventional radiology, interest is growing in the potential benefits of computer-based simulators. Overall, the expected benefit is a reduction of time: for training, operation and hospitalization; but also to promote new surgical devices and techniques. Such expectations are heightened in the intravascular treatment of brain aneurysms for which highly skilled practitioners are required. Recent works [1] attest the feasibility of real-time simulators for such procedures. However, a key requirement is the availability of the blood vessel surface. We propose in this paper an original blood vessel modeling algorithm, specifically suited to real-time simulation of the interventional gesture.

Vessel lumen segmentation can rely on a vast diversity of models [2]. Here, a catheter, or guide, is pushed through the vessels, interacting with their surface. The vessel surface model should enable a smooth motion of the tools and a precise and efficient collision detection. Implicit surface representations, where the surface is defined as the zero-level set of a known function f, are arguably well suited. First, C1-continuous models (with C0-continuous normal) allows for much smoother sliding contacts. Unwanted friction may occur with polyhedral surfaces or level-sets defined on a discrete grid. Second, implicit surfaces offer an improved collision management over parametric surfaces. Indeed, the implicit function value at a point tells whether this point is inside or outside the surface, detecting a collision in the latter case. Furthermore, the implicit function gradient gives a natural direction for the contact force used to handle this collision.

Implicit vessel modeling has been studied, in particular in the context of visualization [3]. A major problem is to handle unwanted blending between two

structures that are geometrically close, but topologically far from each other. Locally adapting the blending [4] has proven efficient, but is not a valid answer in the context of simulation because the simulated tool motion is discrete. The inclusion test might therefore remain successful even though a crossing over a small gap occurred during the time step. Since the seminal work of Masutani [5] the need for local models, based on the centerline of the vascular tree, has been recognized. Placing a graphics primitive at each point of the centerline [6], or using convolution [3] lacks precision where the vessel section is not circular (or elliptic), and does not correctly handle pathologies such as aneurysms. Schumann [7] addressed both problems and used Multi-level Partition of Unity (MPU) implicits, to get a locally defined model with large and general modeling capabilities. However, the blending issue is not addressed and the model is not related to the topology. Furthermore, MPU implicit gradient gives an appropriate contact direction close to the surface but could mislead contact forces elsewhere.

The proposed approach reconstructs the vessel surface as a tree of local implicit models: one local implicit model is placed at each point on the vessel centerline. Complex branching shapes such as the brain vasculature can thus be modeled in a way that is very efficient for interactive simulation. Implicit surfaces generated by skeletons were chosen for their locality and their genericity. Model fitting is defined as an energy minimization problem, alternating with a model refinement as in [8], but improving over this latter work. Our complete modeling algorithm is described in Section 2. Section 3 reports our results on a set of 5 3D rotational angiography (3DRA) patient data. Finally, we discuss our results and give a conclusion of our work in Section 4.

2 Modeling algorithm

Our algorithm requires that the vessel centerline is extracted as a tree, not necessarily dense, and that a local vessel radius estimate is available at each point on this centerline. Many valuable methods can provide this initial structure [2]. Moreover, a set of points, located on the local vessel surface, should be associated with each point on the centerline. In practice, the results presented in Section 3 rely on [9]. However, should the vessel surface be available as a mesh, selecting the vertices in the vicinity of each point on the centerline should also provide such a point set. In this case, our method could be seen as a mesh implicitization. Our overall idea is to fit each local point set with an implicit surface generated by a point-set skeleton (Section 2.1). The local models are organized under the same tree structure as the point centerline (Section 2.2). Solving contact constraints at each point of the interventional tool, is performed using the appropriate local implicit model as the vessel surface (Section 2.3).

2.1 Local Implicit Modeling

Implicit formulation An implicit iso-surface generated by a point-set skeleton is expressed as the zero-level set of a function f, a sum of implicit spheres: $f(X;p) = T - \sum_{j=1}^{N_b} \alpha_j \phi\left(\frac{|X-C_j|}{\rho_j}\right)$.

T is the isosurface threshold, $\{\alpha_j\}$ are positive weights, and $\{C_j\}$ is the point set skeleton. Each implicit sphere $\#j$ is defined by a symmetric spherical function, centered on C_j, of width ρ_j. The local field function, or kernel, is a function $\phi : \mathbb{R} \to \mathbb{R}^+$, rapidly decreasing to 0 at infinity. For example, all results of Section 3 were produced using the 'Cauchy' kernel [10]: $\phi(x) = (1 + x^2/5)^{-2}$ (dividing factor 5 normalizes the kernel such that $\phi''(1) = 0$).

Such objects were called differently depending on the kernel used [10]. Our method is not kernel-dependent, and was successfully used with the computationally less efficient Gaussian kernel. Muraki [11] was the first to use this type of model in the context of object reconstruction. Following this seminal work, we shall use the terms *blob* for an implicit sphere, and *blobby models* as a generic name for the implicit models.

In our particular simulation context, in order to help predict collisions, and have the function give a valid contact force direction, the algebraic value $f(X; p)$ at point X should relate monotonously to the geometric distance of X to the surface. We set $\alpha_j = \rho_j$, which obviously establishes the sought relation in the case of a single blob. Meanwhile, redundancy in the parameters of f is dismissed.

Energy Formulation Fitting a surface to N points $\{P_i\}_{1 \leq i \leq N_p}$ can be written as an energy minimization problem [8, 11]. We propose to combine 3 energy terms: $\mathcal{E} = \mathcal{E}_d + \alpha\mathcal{E}_c + \beta\mathcal{E}_a$ where $(\alpha, \beta) \in {\mathbb{R}^+}^2$, and:

- $\mathcal{E}_d = 1/N_p \sum_i f(P_i; p)^2$
  translates the algebraic relation between data points and the zero-level set. It gives a raw expression of the approximation problem.
- $\mathcal{E}_c = 1/\left(N_b(N_b-1)\right) \sum_{j \neq k} \left(\frac{s\sqrt{\rho_j\rho_k}}{|C_k - C_j|}\right)^{12} - 2\left(\frac{s\sqrt{\rho_j\rho_k}}{|C_k - C_j|}\right)^{6}$
  is Lennard-Jones energy. Each term is minimal (with value -1) for $|C_j - C_k| = s\sqrt{\rho_j\rho_k}$, being repulsive for blobs closer than this distance, and attractive for blobs further away. It imposes some cohesion between neighboring blobs to avoid leakage where data points are missing, while preventing blobs from accumulating within the model.
- $\mathcal{E}_a = 1/N_p \sum_i \kappa(P_i)^2$
  $\kappa(P)$ is the mean curvature. It can be computed in a closed form at any point in space from the implicit formulation [12]

$$\kappa(P) = \frac{\nabla f^t H_f \nabla f - |\nabla f|^2 trace(H_f)}{2|\nabla f|^3}$$

  where ∇f is the implicit function gradient and H_f its Hessian matrix, both computed at point P. This energy smoothes the surface according to the minimal area criterion. In particular, the wavy effect that could stem from modeling a tubular shape with implicit spheres, is reduced.

Behind the rather classical form given above for the energy terms, it is important to notice that the whole energy is known under a closed-form expression. As a consequence, closed-form expressions were derived for its gradients with respect to the blobby model paramaters $\{\rho_j\}$ and $\{C_j\}$.

Selection-Subdivision The blob subdivision procedure proposed in the seminal work [11] was exhaustive and time consuming. A blob selection mechanism was added in [8], measuring the contribution of each blob to $\mathcal{E}_d$ in a user-defined window, and choosing the main contributor. User input is not an option in our context where thousands of blobby models are handled (see Section 3. Moreover, we experimentally noted that this technique was prone to favor small blobs, thus focusing on details, before dealing with areas roughly approximated by one large blob. This behavior is caused by this selection mechanism using the algebraic distance to the implicit surface. Our criterion relies upon the geometric distance approximation proposed by [13]. Point P_{i^*} farthest to the surface is such that:

$$i^* = \arg \max_{1 \leq i \leq N_p} \frac{|f(P_i; p)|}{|\nabla f(P_i; p)|} \tag{1}$$

The blob $\#j^*$ whose isosurface is the closest to P_{i^*} is selected (according to Taubin's distance). Note that this criterion is valid in large areas because we set $\alpha_j = \rho_j$ in the definition of f. The subdivision step then replaces this blob with two new ones. Their width ρ'_{j^*} is chosen such that two blobs, centered on C_{j^*}, of width ρ'_{j^*} would have the same isosurface as one blob centered on C_{j^*}, with width ρ_{j^*} (the formula depends on the kernel). The first new blob is centered on C_{j^*}, while the second is translated by $\rho_{j^*}/10$ towards P_{i^*}.

Optimization Such a gradual subdivision procedure may lead to a dramatic increase in the number of blobs, and hence the size of the optimization problem. The locality of the kernel ϕ allows us to focus the optimization onto the newly created pair of blobs. More exactly, only the new blob that is slightly misplaced is optimized, the other blobs remaining constant. The energy is minimized using Polak-Ribiere conjugate gradient (PR) algorithm, taking advantage of the closed-form expressions of both the energy and its gradients. A single minimization loop consists in one PR minimization over the center (3 variables), followed by one on the width (1 variable). In practice, a maximum of 5 loops proved sufficient.

2.2 Global Implicit Modeling

The algorithm is initialized by placing two blobs at each node on the centerline that form an elongated shape along the vessel direction, with the same diameter as the vessel width. The local data point sets are concatenated so as to overlap with their neighbors, and generate smooth transition between adjacent blobby models. The following three steps are then applied on each blobby model:

1. A first energy minimization is performed over the full blobby model, with a single minimization loop (all centers then all widths of its blobs).
2. The subdivision process is applied, as described above. The process is stopped when a maximum number N_s of subdivisions is reached, or the distance between P_{i^*} and blob $\#j^*$ drops below a threshold t_g.
3. The model is fine tuned by a single energy minimization loop over the full blobby model, as in step 1.

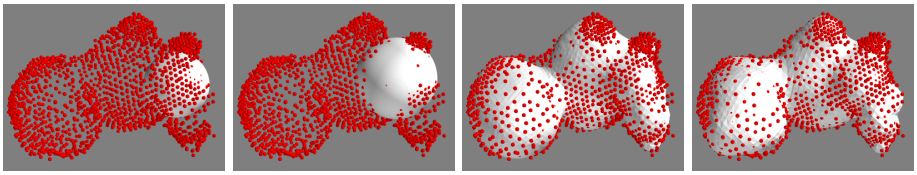

Fig. 1. Implicit modeling of an aneurysm. The points $\{P_i\}$ are in red. (From left to right) Initialization with a single blob ; after the first minimization ; after 25 subdivisions ; final result (100 subdivisions)

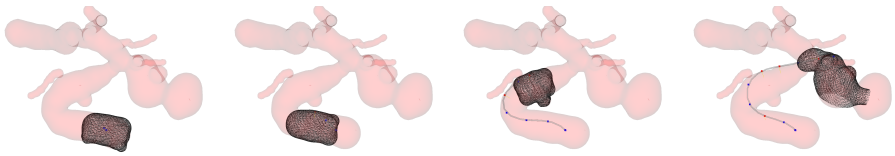

Fig. 2. Selection of a local blobby model during simulation. The current local blobby model surface used to solve the constraints at the tool tip is displayed in wireframe for 4 simulation steps. The overall vessel surface is shown in transparent red. Discrepancies with the blobby models might occur, due to this surface being simplified in order to minimize the CPU load devoted to visualization.

Redundant blobs are either ejected far away from the surface during the optimization over the centers, or their width is reduced to almost zero during the minimization over the widths. A simple clean-up procedure was applied after each minimization loop: blobs further than 20 mm from the node (twice the diameter of the largest artery), and blobs whose width was below 0.02 mm (10% of the voxel size) were removed from the blobby model. In combination with the subdivision process, this clean-up enables the algorithm to simulate large blob displacements, which is hardly possible with the local minimization, thereby adding robustness to poor initialization. Figure 1 illustrates the essential steps of this fitting algorithm and its capacity to model complex shapes, such as aneurysms, even from rough initialization.

Note that in this algorithm, each local blobby model is treated independently from the others, enabling a parallel computation.

2.3 Using the local models for simulation

Most of the interventional tools are slender and their motion can be defined by that of longitudinal nodes. During a simulation step, contact with the vessel surface is detected and solved for each point on the tool. Each tool point is thus linked to its closest point on the centerline and the associated local blobby model is used as the surface constraint. In order to take into account the topology, only

the neighbors of the current centerline point are considered as candidates to update the surface constraint during the motion of the tool (see Fig. 2).

Our model is adapted to simulation, but not to visualization. We translated the above model selection procedure to give a visual impression of the result in the next section: each local isosurface was first extracted with marching cubes, and was then cut by the median planes separating the current centerline point from each of its neighbors. No blending was performed between adjacent blobby models. The final surface presents as a stack of individual surfaces.

3 Results

3.1 Experimental setup

A set of 5 patient data was used for validation. Each patient data set consisted of a 3DRA acquired on a vascular C-arm (Innova 4100, GE Healthcare) during the intra-arterial injection of the internal carotid artery. Each 3DRA volume is a 512^3 isotropic voxel cube, between 0.18-0.22 mm voxel size. For each patient, the vessels were tracked and points were extracted at their surface using [9].

The implicit modeling procedure was applied to all 5 datasets. The algorithm parameters were tuned on the carotid artery of one patient, and thereafter used for all patients. The 'Cauchy' kernel was chosen for its computational efficiency. The energy weights were tuned so the energy values are of the same order of magnitude, leading to $\alpha = 10^{-5}$ and $\beta = 10^{-3}$. For Lennard-Jones energy, s was set to 2 (natural distance between two blobs is twice the geometric mean of their width). The isolevel value T was set to 0.1. Indeed, since we set the weight of each blob equal to its width, blobs whose width is below T do not generate any isosurface (maximum function value is $\rho_j < T$). We therefore chose T to be approximately half the voxel size.

3.2 Geometric precision assessment

The geometric error of fit of a local blobby model with respect to a point set was estimated using Taubin's approximate distance. When measured at each data point, it provides a set of statistical samples for this error: the error of fit for a blobby model, d_{bm} was defined as the 90th percentile on this set, which is a more strict but also more robust measure than the mean. Table 1 reports the distribution, in percents, of d_{bm} measured on all 42840 blobby models. Four classes were considered: below 0.5, 1, 2 and over twice the voxel size (which varies between 0.18 and 0.22mm). Two parameters in particular have an impact on the geometric precision: the distance threshold t_g, expressing a targetted accuracy of fit, and the maximum number N_s of subdivision, expressing the maximum complexity allowed for the models. Measurements were performed for $t_g = 0.5, 0.3$ and 0.2 mm, $N_s = 100, 50, 30$ giving 9 algorithm configurations.

At most 1% of blobby models had an error of fit above 1 voxel. t_g appeared as the only parameter to impact the result and $t_g = 0.3$mm was the only case where

t_g	0.5mm			0.3mm			0.2mm		
N_s	30	50	100	30	50	100	30	50	100
$d_{bm} < 0.5$	47.44	47.18	47.29	60.28	65.07	70.52	89.86	90.07	90.14
$0.5 < d_{bm} < 1$	51.64	51.82	51.83	39.17	34.53	29.11	9.75	9.60	9.57
$1 < d_{bm} < 1.5$	0.85	0.96	0.83	0.48	0.34	0.32	0.32	0.26	0.23
$2 < d_{bm}$	0.06	0.04	0.04	0.06	0.06	0.05	0.08	0.07	0.06

Table 1. Distribution in % of the blobby models according to the error of fit (d_{bm}, in voxels) in 4 classes (left column). 9 algorithm configurations were investigated depending on parameters t_g (targetted accuracy of fit) and N_s (maximum model complexity).

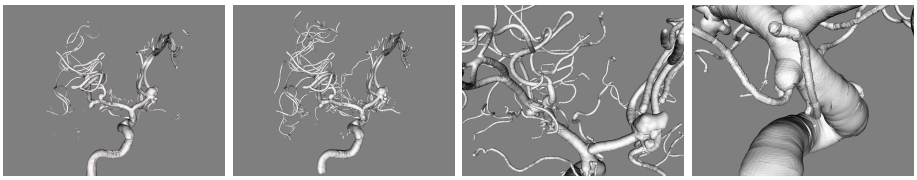

Fig. 3. Visible assessment (from left to right): isosurface from the raw data set ; isosurface from the blobby models (13956 models, 51272 blobs, $t_g = 0.3, N_s = 100$): much more vessels are visible; close-up showing smooth transitions between models ; close-up on a small artery branching onto the carotid: the connection is difficult to model

N_s had a noticeable influence. In our opinion, N_s only has an influence when t_g is adapted to the details to be modeled in the data: $t_g = 0.5$mm was too rough and $t_g = 0.2$mm led the blobby models to fit noisy data points. $t_g = 0.3$mm and $N_s = 100$ provided the best algorithm configuration.

3.3 Model efficiency and computation time

The computation time to model one patient was between 15 and 30 minutes, depending on the number of nodes in the vascular tree (between 5461 and 13956 nodes). A sample result is shown on Figure 3, demonstrating a very good vessel resolution and smooth transitions between neighboring blobby models, except at some very isolated locations. Discontinuities between models do not depend on the kernel, but are mostly sensitive to inhomogeneities in the point set density. Despite discontinuities are below voxel size, we believe they prohibit using our model for blood flow simulation using Computational Fluid Dynamics (CFD). Therefore, direct reconstruction from image data is currently being investigated.

A rule of thumb to measure the efficiency of our model, is to count the number of primitives used. For each patient we compared the total number of blobs (with $t_g = 0.3$ and $N_s = 100$) to the number of triangles in a vessel isosurface mesh of a similar visual quality (see e.g. Fig. 3, left). The ratio #triangles/#blobs ranged from 7 to 13.8, with an average of 10.5. On a more local basis, the average number of blobs per local blobby model was 4.2 (min=3.7, max=4.6).

4 Conclusion

We have presented a patient-specific blood vessel surface reconstruction method that is particularly suited to the simulation of interventional radiology procedures. The final model presents as a tree of implicit surfaces generated by local skeletons. Each surface has a limited area of influence which overlaps with its topological neighbors to provide a smooth transition during the tool guidance. The model fitting is guided by the minimization of an energy, with a closed-form expression, alternating with an efficient blob selection and subdivision scheme. The reconstructed models provided a sub-voxel approximation of the vessel surface on 5 patient data. This model was integrated in our simulation software and preliminary tests proved the adequacy and efficiency of our model in that context. As a perspective, we aim at reconstructing directly from image data.

Acknowledgments

The authors are grateful to Prof. R. Anxionnat (CHU Nancy) for the data.

References

1. Dequidt, J., Duriez, C., Cotin, S., et al.: Towards interactive planning of coil embolization in brain aneurysms. In: MICCAI 2009. Volume 5761 of LNCS. (2009) 377–385
2. Lesage, D., Angelini, E., Bloch, I., et al.: A review of 3D vessel lumen segmentation techniques: Models, features and extraction schemes. MedIA **13** (2009) 819–845
3. Preim, B., Oeltze, S.: 3d visualization of vasculature: An overview. In: Visualization in Medicine and Life Sciences. Mathematics and Visualization. (2008) 39–59
4. Bernhardt, A., Barthe, L., Cani, M.P., et al.: Implicit blending revisited. Comput. Graph. Forum **29** (2010) 367–375
5. Masutani, Y., Masamune, K., Dohi, T.: Region-growing based feature extraction algorithm for tree-like objects. In: Visualization in Biomedical Computing. Volume 1131 of LNCS. (1996) 159–171
6. Tyrrell, J., di Tomaso, E., Fuja, D., et al.: Robust 3-D modeling of vasculature imagery using superellipsoids. IEEE Trans. Med. Imag. **26** (2007) 223–237
7. Schumann, C., Neugebauer, M., Bade, R., et al.: Implicit vessel surface reconstruction for visualization and CFD simulation. IJCARS **2** (2008) 275–286
8. Tsingos, N., Bittar, E., Cani, M.P.: Implicit surfaces for semi-automatic medical organ reconstruction. In: Computer Graphics Internat. (CGI'95). (1995) 3–15
9. Yureidini, A., Kerrien, E., Cotin, S.: Robust RANSAC-based blood vessel segmentation. In: SPIE Medical Imaging. Volume 8314., SPIE (2012) 83141M
10. Sherstyuk, A.: Kernel functions in convolution surfaces: A comparative analysis. The Visual Computer **15** (1999) 171–182
11. Muraki, S.: Volumetric shape description of range data using Blobby Model. SIGGRAPH Comput. Graph. **25** (1991) 227–235
12. Goldman, R.: Curvature formulas for implicit curves and surfaces. Computer Aided Geometric Design **22** (2005) 632–658
13. Taubin, G.: Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. PAMI **13** (1991) 1115–1138