5 Modeling neurons III: 3D modeling and simulation

In this chapter we will extend our modeling approaches to higher-dimensional, i.e. 2D and, in particular 3D, methods. Whenever biological processes need to include a full-scale representation of the intra- and extracellular space and/or the detailed 3D morphology needs to be incorporated in numerical simulations (e.g., when studying structure-function interplay), partial differential equations will need to be solved on realistic computational domains. The following sections will therefore discuss methods of image reconstruction to get from raw microscopy data to surface triangulations of the computational boundaries and methods for discretizing and solving models based on partial differential equations.

5.1 3D image reconstruction

Two main issues need to be addressed: (i) background noise and (ii) missing structural information in raw microscopy data. Image filters can reduce noise and smooth structure in order to extract morphological information in the form of surface grid triangulations.

Image filters

Filters smooth raw data and create similarity in gray values, belonging to the structure. Let G_j be the gray value of voxel j.

Examples:
(A) Median filter Let F_M be the median filter function

$$F_M(G_j) = \frac{1}{k} \cdot \sum_{i=1}^{k} G_i \tag{5.1}$$

(B) Gaussian filter Including a Gaussian weighting function $g(x_i, x_j)$ with voxel positions x_i and x_j yields

$$\begin{aligned} F_G(G_j) &= \sum g(x_j, x_i) \cdot G_i \\ \text{with} \quad g(j,i) &:= \frac{1}{2\pi\sigma^2} \cdot e^{-\frac{1}{2}\left(\frac{x_i^2 + x_j^2}{\sigma^2}\right)} \end{aligned} \tag{5.2}$$

The drawback of these *isotropic* filters is that they do not respect structural information and thus filigreed cellular structures could be altered significantly. We extend our list of demands to include *smoothing* and *structure conservation*.

Diffusion filters

The idea behind diffusion filters is to interpret the gray scale image as a discrete distribution of mass, where each gray value represents the mass/concentration inside the corresponding voxel. Based on this assumption we can consider a physical diffusion process on the image data set.

Diffusion

1. *Fick's law* states:

$$\begin{aligned} &\text{Flux}\, \Phi = -\nabla u \\ &\nabla : \text{ Gradient} \begin{pmatrix} \partial/\partial x \\ \partial/\partial y \\ \partial/\partial z \end{pmatrix} \text{in 3D} \end{aligned} \tag{5.3}$$

2. *Conservation of mass*: Let V be an arbitrary volume element and u the concentration in V. Then the concentration changes within V are given by

$$\int_V \frac{\partial u}{\partial t} d\vec{x}.$$

Given that no sources or sinks are present in V, the changes in u are directly related to the flux across the boundary ∂V of V:

$$-\int_{\partial V} \Phi \cdot \vec{n}\, ds = \int_V \frac{\partial u}{\partial t} d\vec{x} \tag{5.4}$$

Using the Gauss theorem

$$\int_V div\, F(\vec{x}) d\vec{x} = \int_{\partial N} F(\vec{x}) \cdot \vec{n}\, ds, \tag{5.5}$$

we get

$$-\int_{\partial V} \Phi \cdot \vec{n}\, ds = -\int_V div\, \Phi\, d\vec{x} = \int_V \frac{\partial u}{\partial t} d\vec{x} \tag{5.6}$$

$$\Rightarrow -div\, \Phi = \frac{\partial u}{\partial t} \tag{5.7}$$

$$\Rightarrow \frac{\partial u}{\partial t} = -div\, \nabla u \tag{5.8}$$

In general we can add a diffusion tensor D, such that

$$\frac{\partial u}{\partial t} = -div(D\nabla u). \tag{5.9}$$

If D is scalar, we get

$$\frac{\partial u}{\partial t} = D \cdot \Delta u \text{ with Laplace operator } \Delta; \quad \Delta := div\,\nabla = \sum_{i=1}^{n} \frac{\partial^2}{\partial {x_i}^2} \tag{5.10}$$

The choice of D allows us to produce different diffusion filter properties.

Examples:

$$\text{(A)} \qquad D = \begin{pmatrix} 1 & & 0 \\ & 1 & \\ 0 & & 1 \end{pmatrix} \Rightarrow \text{isotropic diffusion (e.g., Gaussian blur)} \tag{5.11}$$

$$\text{(B)} \qquad D = \begin{pmatrix} 5 & & \\ & 1 & \\ & & 1 \end{pmatrix} \Rightarrow \text{anisotropic diffusion} \tag{5.12}$$

5.1.1 Choosing diffusion tensor D

Identifying the image set as a distribution of mass now lets us define the quantities *mass*, *center of mass*, and *moments of inertia*. The inertia tensor can be used to identify the orientation of an object in 3D space and whether it is linear, planar, or isotropic. This information, coming from the inertia tensor, can guide the design of the diffusion tensor.

$$\text{Mass:} \qquad M := \sum_i m_i$$

$$\text{Center of mass:} \qquad R := \frac{1}{M} \sum m_i \vec{r}_i \qquad \vec{r}_i : \text{position vector of } x_i$$

Inertia tensor
The inertia tensor, in the discrete setting, is defined by

$$J := \sum_i m_i \begin{pmatrix} y_i^2 + z_i^2 & -x_i y_i & -x_i z_i \\ -y_i x_i & x_i^2 + z_i^2 & -y_i z_i \\ -z_i x_i & -z_i y_i & x_i^2 + y_i^2 \end{pmatrix}, \tag{5.13}$$

where m_i is the mass (gray value) and (x_i, y_i, z_i) is the position of voxel i. Since J is symmetric and positive definite:

$$\Rightarrow \quad \exists\, \lambda_1, \lambda_2, \lambda_3 \text{ and } V_1, V_2, V_3 \text{ with}$$
$$J \cdot V = \lambda \cdot V \text{ and main axis transformation yields}$$
$$J = (V_1 V_2 V_3) \underbrace{\begin{pmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \lambda_3 \end{pmatrix}}_{\text{eigenvalues of J}} (V_1 V_2 V_3)$$

Eigenvalue distribution carries information about the type of object:

1. $\lambda_1 \ll \lambda_2 \approx \lambda_3$: This indicates a linear structure
2. $\lambda_1 \approx \lambda_2 \ll \lambda_3$: This indicates a planar structure
3. $\lambda_1 \approx \lambda_2 \approx \lambda_3$: This indicates an isotropic structure

5.1.2 The inertia based diffusion filter

Using this structural information, we can choose the diffusion tensor D to allow diffusion along the relevant directions, i.e., in 1. along V_1, in 2. along V_1, V_2, and in 3. along V_1, V_2, V_3. For dendritic structures this would yield

$$\begin{pmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \lambda_3 \end{pmatrix} \rightarrow \begin{pmatrix} \epsilon & & \\ & 1 & \\ & & 1 \end{pmatrix} =: D_L \tag{5.14}$$

For two-dimensional membranes we get

$$\begin{pmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \lambda_3 \end{pmatrix} \rightarrow \begin{pmatrix} \epsilon & & \\ & \epsilon & \\ & & 1 \end{pmatrix} =: D_P \tag{5.15}$$

Back to the diffusion equation

$$\frac{\partial u}{\partial t} = div(D \nabla u) \quad \text{in } \Omega$$
$$\text{Set } D = \begin{cases} D_L & \text{for linear structures, like dendrites and axons} \\ D_P & \text{for planar membranes, like soma or organelles} \end{cases}$$

This approach lead us to an inertia-based, anisotropic diffusion filter. A filter has the function of creating voxel similarities between those representing the structure of interest and to diminish background noise. From processed image data one can then generate a surface triangulation using, e.g., a marching cubes algorithm.