Robust membrane detection based on tensor voting
for electron tomography

Antonio Martinez-Sanchez[a], Inmaculada Garcia[b], Shoh Asano[c], Vladan Lucic[c],
Jose-Jesus Fernandez[d,*]

[a]*Supercomputing and algorithms group. Associated Unit CSIC-UAL, Universidad de Almeria,
04120 Almeria, Spain*
[b]*Supercomputing and algorithms group. Dept. Computer architecture. Universidad de Malaga,
29080 Malaga, Spain*
[c]*Max-Planck Institute of Biochemistry, Am Klopferspitz 18, 82152 Martinsried, Germany*
[d]*National Centre for Biotechnology. National Research Council (CNB-CSIC). Campus UAM.
Darwin 3. Cantoblanco. 28049 Madrid. Spain.*

[*]Corresponding author:
National Centre for Biotechnology.
National Research Council (CNB-CSIC).
Campus UAM. C/ Darwin 3. Cantoblanco.
28049 Madrid. Spain.
Tel: +34 91 585 4619. Fax: +34 91 585 4506
Email: jj.fernandez@csic.es

Abstract

Electron tomography enables three-dimensional (3D) visualization
and analysis of the subcellular architecture at a resolution of a few
nanometres. Segmentation of structural components present in 3D
images (tomograms) is often necessary for their interpretation. How-
ever, it is severely hampered by a number of factors that are inherent
to electron tomography (e.g. noise, low contrast, distortion). Thus,
there is a need for new and improved computational methods to fa-
cilitate this challenging task. In this work, we present a new method
for membrane segmentation that is based on anisotropic propagation
of the local structural information using the Tensor Voting algorithm.
The local structure at each voxel is then refined according to the in-
formation received from other voxels. Because voxels belonging to
the same membrane have coherent structural information, the under-
lying global structure is strengthened. In this way, local information
is easily integrated at a global scale to yield segmented structures.
This method performs well under low signal-to-noise ratio typically
found in tomograms of vitrified samples under cryo-tomography con-
ditions and can bridge gaps present on membranes. The performance
of the method is demonstrated by applications to tomograms of differ-
ent biological samples and by quantitative comparison with standard
template matching procedure.

Keywords

3

1. Introduction

Electron tomography (ET) enables three-dimensional (3D) visualization and analysis of the subcellular architecture and macromolecular organization of cells and tissues in situ at a resolution of a few nanometers (Lucic et al., 2013). This technique involves the acquisition of electron microscopy projection images of a specimen at different orientations. These images are then combined by means of tomographic reconstruction methods to yield the 3D volume (Fernandez, 2012).

Segmentation of the 3D volume into its constitutive structural elements is key for their interpretation. However, it proves to be challenging because of a number of factors such as the crowded cellular environment, the distortion caused by the missing wedge and noise, which is particularly high in ET of fully hydrated and vitrified samples (cryo-ET) (Volkmann, 2010; Fernandez, 2012). Thus, segmentation constitutes a major bottleneck in ET, especially in those studies intended to visualize the subcellular architecture under cryo-conditions. Although several computational segmentation methods have been presented, none has shown general applicability yet. As a consequence, manual segmentation is still a method of choice.

Software packages often used in ET have been gradually including segmentation procedures based on the most known computational techniques (Watershed transform and thresholding (Volkmann, 2002; Cyrklaff et al., 2005)). This makes segmentation a semi-automatic process, thus facilitating its use. In the last several years, there have been significant advances towards computational detection of specific structural features within tomograms, like membranous structures, filaments and microtubules (Lebbink et al., 2007; Sandberg and Brega, 2007; Moussavi et al., 2010; Nurgaliev et al., 2010; Rigort et al., 2012; Weber

4

et al., 2012). Many of these methods rely on some sort of template matching. That is, they search for a template that is suited to the targeted feature, typically by means of cross-correlation techniques. In contrast to these methods, we have recently developed a differential geometry-based segmentation that is particularly suited for membranes (Martinez-Sanchez et al., 2011, 2013). We first proposed a method that uses a local membrane detector based on the Hessian tensor (Martinez-Sanchez et al., 2011). Later, we improved this detector and extended the abilities of the framework to characterize and classify the detected membranous structures (Martinez-Sanchez et al., 2013). Nevertheless, this local membrane detector still presented several limitations. Namely, gaps that may appear on membranes due to experimental imaging conditions were not properly filled. Also, membrane-attached structures were not discarded and were instead segmented as part of the membranes. A non-trivial postprocessing stage was required in these cases to actually extract the membrane voxels. Finally, the detector used was suitable for membranes with ridge-like (i.e. local maximum) profile, and was therefore unable to identify edge-like structures (e.g. membrane of vesicles having dense interior).

In this work, we propose a more robust local membrane detector. The method is based on broadcasting differential information through the 3D space using the Tensor Voting algorithm (Tong et al., 2004). In this way, nearby voxels that belong to the same membranous structure enhance each other's structural information. As a result, the new local detector can fill the gaps present on membranes, disregards structures apposed to the membranes, and it is more robust against low signal-to-noise ratio, thus simplifying the postprocessing stage. In this work we also show that membranes having ridge-like and edge-like profiles can be detected using the

5

same procedure but different tensor that provides the differential information.

2. Background on local membrane detection

2.1. Hessian tensor-based detection

It is assumed that, at a local scale, a membrane can be modelled as a plane-like structure with membrane density profile (in the direction perpendicular to the membrane) following a Gaussian function. That is, the membrane profile is ridge-like, its density decreases as a function of the distance from the center of the membrane.

Membrane detection starts with the application of a scale-space operation on a greyscale volume, typically implemented as Gaussian low-pass filtering. This step is used to isolate the information at a given scale σ, thus filtering out noise and all features smaller than the scale. An additional benefit of this step is that scale-space can smooth membranes making their profile closer to Gaussian. Typically, σ is set to the thickness (expressed in voxels) of the targeted membrane (Martinez-Sanchez et al., 2011).

Previously, our local detector for ridge-like membranes was based on the Hessian tensor (Martinez-Sanchez et al., 2011). This tensor provides information about the second order density variation, as it is defined as:

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 L}{\partial x^2} & \frac{\partial^2 L}{\partial x \partial y} & \frac{\partial^2 L}{\partial x \partial z} \\ \frac{\partial^2 L}{\partial x \partial y} & \frac{\partial^2 L}{\partial y^2} & \frac{\partial^2 L}{\partial y \partial z} \\ \frac{\partial^2 L}{\partial x \partial z} & \frac{\partial^2 L}{\partial y \partial z} & \frac{\partial^2 L}{\partial z^2} \end{bmatrix} \tag{1}$$

where L denotes the volume after the scale-space operation and $\frac{\partial^2 L}{\partial i \partial j} \ \forall i, j \in (x, y, z)$ are its second order partial derivatives. As a result of the eigen-analysis of the

6

Hessian tensor, three orthogonal eigenvectors $\vec{v}_i$ and their corresponding eigenvalues λ_i (representing second order derivatives along $\vec{v}_i$) are obtained, which characterize the local structure around any voxel of the volume. If we assume $|\lambda_1| > |\lambda_2| > |\lambda_3|$, then the first eigenvector $\vec{v}_1$, i.e. the one whose eigenvalue exhibits the largest absolute value, points to the direction of the maximum curvature. If the local structure is a plane, $\vec{v}_1$ points to the direction perpendicular to the plane and the following relationship holds $|\lambda_1| >> |\lambda_2| \approx |\lambda_3|$. This led us earlier to propose a local detector (so-called membrane strength, M) defined as follows (Martinez-Sanchez et al., 2011):

$$M = \begin{cases} \frac{(|\lambda_1| - \sqrt{\lambda_2 \lambda_3})^2}{|\nabla L|^2} & \lambda_1 < 0 \\ 0 & \text{otherwise} \end{cases} \qquad (2)$$

where $|\nabla L|$ denotes the gradient of the volume L resulting from the scale-space operation.

We used two additional steps to detect peaks of M corresponding to membranes. A threshold t_M over M was imposed to select membrane-like voxels. This was then coupled with the non-maximum suppression (NMS) criterion which selects only the ridge points (i.e. the local maxima in the direction perpendicular to the membrane). The use of NMS results in detected membranes being represented as one-voxel thick surfaces in the 3D space. In summary, the local detector for ridge-like membranes is given by the following equation (Martinez-Sanchez et al., 2013), where the first condition represents NMS, δ is a small number and $\mathbf{x} \in \mathbb{R}^3$ denotes a voxel of the volume:

$$\begin{cases} (L(\mathbf{x}) > L(\mathbf{x} - \delta\vec{v}_1)) \text{ and } (L(\mathbf{x}) > L(\mathbf{x} + \delta\vec{v}_1)) \\ M(\mathbf{x}) \geq t_M \end{cases} \qquad (3)$$

7

2.2. Structure tensor to detect edge-like membranes

The detector defined in the previous section can detect ridge-like membranes, but it is not suitable for the detection of membranes with edge-like profile where density on one side of the membrane is similar to that of the membrane (e.g. those presented by a densely filled vesicle). Consequently, a new detector for this type of membranes is required. To this end, we focus our attention to the Structure tensor (Weickert, 1998), also known as the second moment tensor, which is given by:

$$
\mathbf{J} = \begin{bmatrix} \left(\frac{\partial L}{\partial x}\right)^2 & \frac{\partial L}{\partial x}\frac{\partial L}{\partial y} & \frac{\partial L}{\partial x}\frac{\partial L}{\partial z} \\ \frac{\partial L}{\partial x}\frac{\partial L}{\partial y} & \left(\frac{\partial L}{\partial y}\right)^2 & \frac{\partial L}{\partial y}\frac{\partial L}{\partial z} \\ \frac{\partial L}{\partial x}\frac{\partial L}{\partial z} & \frac{\partial L}{\partial y}\frac{\partial L}{\partial z} & \left(\frac{\partial L}{\partial z}\right)^2 \end{bmatrix} \tag{4}
$$

where $\frac{\partial L}{\partial i}$ $\forall i \in (x, y, z)$ are the first order derivatives of the scale-spaced volume L. The eigen-analysis of this tensor proceeds as in the previous case (Fernandez and Li, 2003, 2005). The first eigenvector $\vec{v_1}$, i.e. the one whose eigenvalue exhibits the largest value, points to the direction of the maximum variation. Also, a local plane satisfies $|\lambda_1| >> |\lambda_2| \approx |\lambda_3|$. However, the Hessian and Structure tensors differ regarding the exact position of the detected surface. For a ridge profile L, the local maximum of the largest eigenvalue ($|\lambda_1|$) of the Hessian tensor corresponds to the maximum of L (Figure 1(Left)). For an edge profile L, the local maximum of the largest eigenvalue ($|\lambda_1|$) of the Structure tensor corresponds to the inflection point of L (Figure 1(Right)). Consequently, the Structure tensor is well suited to detect edge-like local structures. Moreover, the output of this edge detector is equivalent to the output of the Hessian-based detector (red curves in Figure 1). This means that the eigen-analysis of the Structure tensor allows detection of edges and their conversion into ridges. This, in turn, enables appli-

8

cation of all our ridge-based methodology to detect membranes with edge profile. Therefore, the ridge-like profiles detected by the Hessian-based detector and the edge-like profiles detected by the Structure tensor-based detector can be further processed in the same way.

In this work, first and second order derivatives required for the components of the tensors have been implemented based on central differences (Frangakis and Hegerl, 2001).

3. Tensor Voting

In this section, the theoretical background on Tensor Voting (TV) is presented. Our aim is to apply TV to enhance surface- or plane-like structures, such as membranes in tomograms. In order to benefit from an efficient implementation of the TV algorithm in 2D, and because of the high computational demands presented by TV in 3D we chose to apply TV on tomographic slices. This is supported by the fact that membranes in tomograms appear as curves in 2D planes (Martinez-Sanchez et al., 2011). Thus, we proceed by using TV to detect curves on the slices. In our procedure, TV is applied to all slices along all three major axes. Therefore, in this section 2D TV algorithm and an efficient implementation of this algorithm are presented.

3.1. Introduction to Tensor Voting

Tensor Voting is a technique used in computer vision for robust identification of salient features, which exhibits excellent performance under particularly challenging noise conditions (Medioni et al., 2000). In TV, voxels propagate information about their local structural features across their neighbourhood. The local properties at each voxel are then refined according to the information re-

9

ceived from the neighbours. The aim is that voxels that belong to the same global feature (e.g. a surface or membrane in our case, which are represented as curves in 2D), and thus have coherent structural information, strengthen each other's contribution to the underlying global structure.

There are several important concepts and steps involved in TV: representation of the local structural information, the initialization of that information, the propagation through votes, the gathering of votes and refinement of the local information. They will be described in the following sections.

3.1.1. Data representation

In TV the local structural information at each voxel is represented by a symmetric non-negative definite second order tensor. This tensor allows encoding saliency of different types of geometric structures (curves and blobs in the 2D case) by its eigenvalues (λ_1 and λ_2, with $\lambda_1 \geq \lambda_2 \geq 0$), and the corresponding eigenvectors ($\vec{v}_1$ and $\vec{v}_2$):

$$\mathbf{T} = \begin{bmatrix} t_{xx} & t_{xy} \\ t_{xy} & t_{yy} \end{bmatrix} = \begin{bmatrix} \vec{v}_1 & \vec{v}_2 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} \vec{v}_1 & \vec{v}_2 \end{bmatrix}^T . \tag{5}$$

Geometric structures can be detected from this tensor as described in the previous section for the Structure tensor and Hessian tensor (Fernandez and Li, 2003, 2005; Martinez-Sanchez et al., 2011, 2013), but converted to the 2D case. Thus, voxels belonging to a local curve have $\lambda_1 >> \lambda_2$, with $\vec{v}_1$ in the direction perpendicular to the curve, and the term $\lambda_1 - \lambda_2$ represents the curve saliency. At voxels that do not show a clear structure (blob-like), the two eigenvalues have similar magnitude. The concept of saliency can be understood as the likelihood of a voxel to belong to one of the basic geometric features.

10

It is useful to decompose the tensor in order to reveal better the different types of saliency as follows:

$$\mathbf{T} = \lambda_1 \vec{v}_1 \vec{v}_1^T + \lambda_2 \vec{v}_2 \vec{v}_2^T = (\lambda_1 - \lambda_2) \vec{v}_1 \vec{v}_1^T + \lambda_2 (\vec{v}_1 \vec{v}_1^T + \vec{v}_2 \vec{v}_2^T) \tag{6}$$

In this decomposition, the two terms are called stick and ball components, respectively (Medioni et al., 2000; Tong et al., 2004). In a voxel belonging to a curve, the stick component dominates because of the high saliency $\lambda_1 - \lambda_2$. In the extreme case: $\lambda_1 > 0$ and $\lambda_2 = 0$, so only the stick component remains. Such a tensor is then referred to as stick tensor, and it encodes a pure local curve with saliency λ_1 and the normal vector $\vec{v}_1$.

Saliency of the stick component of a 2D tensor $\mathbf{T}$ and the direction of the corresponding vector with respect to the x-axis can be calculated analytically, as follows (Franken et al., 2006):

$$S(\mathbf{T}) = \lambda_1 - \lambda_2 = \sqrt{\mathrm{tr}(\mathbf{T})^2 - 4\det(\mathbf{T})} = \sqrt{(t_{xx} - t_{yy})^2 + 4t_{xy}^2} \tag{7}$$

$$\alpha(\mathbf{T}) = \arccos(\vec{v}_1 \cdot \hat{e}_x) = \frac{1}{2} \arg(t_{xx} - t_{yy} + 2it_{xy}) \tag{8}$$

Here, $\mathrm{tr}(\mathbf{T})$ and $\det(\mathbf{T})$ denote the trace and determinant of the matrix $\mathbf{T}$, respectively, $\cdot$ is the dot product, i represents the imaginary unit (i.e. $i^2 = -1$) and arg() is the phase of a complex number.

3.1.2. Initialization and generation of the input tensor field

Tensors in TV are typically initialized to isotropic balls (i.e. equal eigenvalues) or estimated from prior knowledge if available (Medioni et al., 2000; Tong et al., 2004). In this work, we are interested in detecting curve-like structures in

11

the 2D tomographic slices. Thus, for each input 2D slice subjected to TV, Hessian or Structure tensors are calculated and the input tensor field is created as the pure stick component of the eigen-decomposition of the tensor field. Namely, the eigenvectors of the input tensor field are $\vec{v}_i$, $\forall i$, while the saliency is $|\lambda_1 - \lambda_2|$ so that:

$$\mathbf{T} = |\lambda_1 - \lambda_2|\vec{v}_1\vec{v}_1^T \tag{9}$$

This defines an input tensor field $\mathbf{T}(\mathbf{x})$, with $\mathbf{x}$ denoting the voxel coordinates, that represents the preliminary estimated likelihood of each voxel in the input slice to belong to a curve. Tensor Voting procedure refines the tensor information at each voxel by propagating the tensors to the neighbourhood, as presented in the following subsections.

3.1.3. Voting mechanism

Voxels communicate among themselves by propagating information between each other by a process called voting. The information passed is a second order tensor called vote, which is sent from a voter voxel to a receiver voxel. In this work, votes are based solely on stick tensors because they carry information about curves. A vote is propagated along an arc of a circle that contains voter $\mathbf{x}'$ and receiver $\mathbf{x}$ and is perpendicular to the vector that defines the orientation of the vote at the voter point (Figure 2(A)). The propagation of the vote from voter $\mathbf{x}'$ to receiver $\mathbf{x}$ induces the transformation of the stick tensor according to the so-called stick voting tensor field defined as follows (Medioni et al., 2000):

$$\mathbf{V}(\mathbf{x} - \mathbf{x}', \alpha(\mathbf{T}(\mathbf{x}'))) = e^{-\left(\frac{s^2 + d\kappa^2}{\sigma_v^2}\right)}\vec{N}\vec{N}^T \tag{10}$$

12

where $s = s(\mathbf{x} - \mathbf{x}') = \frac{l\phi}{\sin\phi}$ is the length of the arc that connects the voter and the receiver, $\kappa = \kappa(\mathbf{x} - \mathbf{x}') = \frac{2\sin\phi}{l}$ the curvature of the osculating circle, d a constant that controls the decay with respect to curvature and σ_v is the length scale of analysis, which determines the effective neighbourhood size (expressed in voxels). Consequently, the exponential term represents the distance-dependent attenuation of the saliency. The orientation of the vote, $\vec{N}$, is obtained by dragging the vector $\vec{v}_1 = [\cos\alpha\,(\mathbf{T}(\mathbf{x}'))\,, \sin\alpha\,(\mathbf{T}(\mathbf{x}'))]$ that determines the orientation of the stick tensor at voter point $\mathbf{x}'$, along the arc from $\mathbf{x}'$ to the receiver point $\mathbf{x}$ (Figure 2(A)): $\vec{N} = \mathbf{R}(2\phi)\vec{v}_1$, with $\mathbf{R}()$ denoting the rotation matrix.

Figure 2(B) depicts the stick voting tensor field, which is the collection of the second order votes cast by a voter located at the origin, whose stick tensor has unit saliency and orientation along the x-axis: $\mathbf{V}(\mathbf{x}, 0)$. The extension of this voting kernel is limited by σ_v (in practice it is about $3 \times \sigma_v$) and $\phi \le 45^\circ$ (Medioni et al., 2000; Tong et al., 2004).

The voting mechanism expressed in Eq. (10) is usually interpreted as follows (Medioni et al., 2000; Tong et al., 2004). To compute the votes cast by a voter $\mathbf{x}'$, the stick voting field $\mathbf{V}(\mathbf{x}, 0)$ is placed at the location of the voter and with its orientation $\alpha = \alpha(\mathbf{T}(\mathbf{x}'))$ (Medioni et al., 2000; Tong et al., 2004), using tensor rotation operations:

$$\mathbf{V}(\mathbf{x} - \mathbf{x}', \alpha) = \mathbf{R}(\alpha)\,\mathbf{V}(\mathbf{R}^{-1}(\alpha)(\mathbf{x} - \mathbf{x}'), 0)\,\mathbf{R}^{-1}(\alpha) \tag{11}$$

Figure 2(C) shows a sketch of this transformation and voting process. Most implementations of tensor voting are based on this interpretation.

13

3.1.4. Vote collection and interpretation of results

The votes received at each voxel **x** are collected and accumulated by tensor addition so as to yield the resulting output tensor $\mathbf{U}(\mathbf{x})$. Thus, the voting process described above can be considered as a tensor convolution of the input tensor field $\mathbf{T}(\mathbf{x})$ with the voting field $\mathbf{V}(\mathbf{x}, \alpha)$:

$$\mathbf{U}(\mathbf{x}) = \sum_{\mathbf{x}' \in \text{Neigh}(\mathbf{x})} S\left(\mathbf{T}(\mathbf{x}')\right) \mathbf{V}(\mathbf{x} - \mathbf{x}', \alpha(\mathbf{T}(\mathbf{x}'))) \tag{12}$$

where $\mathbf{x}'$ represents a voter within the neighborhood of **x**, which has saliency $S(\mathbf{T}(\mathbf{x}'))$ and orientation $\alpha(\mathbf{T}(\mathbf{x}'))$.

After TV, the output tensor field $\mathbf{U}(\mathbf{x})$ is expected to have more coherent structural information. Voxels belonging to the same geometric feature will have strengthened each other and their tensors will have been modified to enhance the underlying global structure. Interpretation of the final tensors is carried out as described above, in Section 3.1.1, leading to a more reliable estimates of curves (i.e. voxels with high curve saliency $\lambda_1 - \lambda_2$, and with the normal to the curve pointed by $\vec{v}_1$).

3.2. Efficient implementation of Tensor Voting with steerable filters

Tensor voting is a computationally demanding procedure. The standard implementation consists of pre-computing and storing the voting field $\mathbf{V}(\mathbf{x}, 0)$ (Eq. (10)). Translation and rotation of the voting field throughout the image space needed for casting votes (Eqs. (11; 12)) is done by interpolation. Alternatively, storing all rotated voting fields would reduce the voting process to simple look-up operations and interpolation, but the memory consumption would be higher. There exists, however, a more efficient implementation that takes advantage of the theory of steerable filters (Franken et al., 2006). The requirement for selective orientation

14

of the voting field is similar to that encountered with scalar oriented filters, the optimal implementation of which is carried in the framework of the theory of steerable filters (Freeman and Adelson, 1991).

In image processing, a steerable filter is a filter that can be oriented in an arbitrary direction just by a linear combination of a finite number of predefined rotations of the filter (so-called basis functions or filters) (Freeman and Adelson, 1991). Rotation of the steerable filter h by an angle α can be shown as:

$$h(\mathbf{x}, \alpha) = \sum_{m=1}^{M} k_m(\alpha) h_m(\mathbf{x}) \tag{13}$$

where $h_m(\mathbf{x})$ are the basis filters, $k_m(\alpha)$ are the linear coefficients that depend on the rotation angle α, m denotes the index of the basis filters and the number of basis filters is M. In practice, in order to reduce the number of the basis filters required to steer the filter, they are taken from the nonzero coefficients of the angular Fourier decomposition of the filter (Freeman and Adelson, 1991). Filtering an image $f(\mathbf{x})$ with the oriented filter $h(\mathbf{x}, \alpha)$ then becomes:

$$f(\mathbf{x}) * h(\mathbf{x}, \alpha) = \sum_{m=1}^{M} k_m(\alpha)(f(\mathbf{x}) * h_m(\mathbf{x})) \tag{14}$$

where $*$ denotes convolution. In other words, an image filtered at an arbitrary direction can be computed by a simple linear combination of the image filtered with M basis filters h_m, with coefficients $k_m(\alpha)$. If M is sufficiently small, this turns out to be a very efficient method for arbitrary oriented filtering of images.

Here we briefly describe the two key aspects of the derivation of steerable approach to 2D TV in order to understand the internals of our implementation. This derivation, together with the mathematical details, were nicely presented in (Franken et al., 2006). First, steerable tensor voting uses a saliency decay function different from the original one (exponential term in Eq. (10)) so as to make

15

the voting field bandlimited (i.e. to have a finite number of nonzero coefficients of the angular Fourier decomposition): $\frac{1}{16} e^{\frac{-r^2}{2\sigma_v^2}} \cos^4 \phi$.

Second, as explained in Section 3.1.1, the tensors used in 2D TV are 2×2 symmetric matrices having three independent components: $\{t_{xx}, t_{xy}, t_{yy}\}$. Tensor rotation (Eq. (11)) can be computed using a similarity transformation (Golub and van Loan, 1996), as thoroughly described in (Franken et al., 2006) for a general case. Furthermore, rotation of a pure stick tensor, as required here for votes, simplifies to the rotation of a single function V_{-2} derived from the components of the tensor (Franken et al., 2006): $V_{-2} = t_{xx} + 2it_{xy} - t_{yy}$. This function allows easy calculation of the saliency and orientation of the tensor, as defined in Eqs. (7) and (8): $S(\mathbf{V}) = |V_{-2}|$ and $\alpha(\mathbf{V}) = \frac{1}{2} \arg(V_{-2})$, respectively.

Thus, the steerable tensor field rotated by an angle α is finally reduced to the rotation of the steerable function V_{-2} (Franken et al., 2006):

$$V_{-2}(\mathbf{x}, \alpha) = \sum_{m=0}^{4} k_m(\alpha) V_m(\mathbf{x}) \tag{15}$$

where k_m are the linear coefficients:

$$k_m(\alpha) = e^{-2i(m-1)\alpha} \tag{16}$$

and $V_m(\mathbf{x})$ are the basis filters given by:

$$V_m(\mathbf{x}) = \gamma_m \, e^{-\frac{x^2+y^2}{2\sigma_v^2}} \left(\frac{x + iy}{\sqrt{x^2 + y^2}} \right)^{2m} , \text{ for } \mathbf{x} = (x, y) \neq (0, 0) \tag{17}$$

where γ_m has constant values: $\{1, 4, 6, 4, 1\}$ for $m = 0 \ldots 4$, respectively.

Therefore, the voting process originally expressed as a tensor convolution (Eq. (12)) is reduced to a scalar steerable convolution that, similarly to Eq. (14),

16

can be expressed as:

$$U_{-2}(\mathbf{x}) = S(\mathbf{T}(\mathbf{x})) * V_{-2}(\mathbf{x}, \alpha(\mathbf{T}(\mathbf{x}))) = \sum_{m=0}^{4} k_m(\alpha(\mathbf{T}(\mathbf{x})))(S(\mathbf{T}(\mathbf{x})) * V_m(\mathbf{x})) \quad (18)$$

From here, calculation of the output saliency and orientation is straightforward, as mentioned above:

$$\begin{aligned} S(\mathbf{U}(\mathbf{x})) &= |U_{-2}(\mathbf{x})| \\ \alpha(\mathbf{U}(\mathbf{x})) &= \tfrac{1}{2}\arg(U_{-2}(\mathbf{x})). \end{aligned} \quad (19)$$

Eqs. (18) and (19) reveal that only five convolutions are required to accomplish TV. Moreover, computation of these convolutions in Fourier speeds up the calculation significantly (Franken et al., 2006). Therefore, this steerable approach provides a very efficient implementation of TV, well suited to deal with large tomograms commonly encountered in ET.

The implementation of steerable 2D TV can then be summarised as follows:

- **Initialization of the input tensor field**.

  The saliency and orientation of the input tensor field, ($S_{\text{in}}(\mathbf{x})$ and $\alpha_{\text{in}}(\mathbf{x})$) are initialized based on the eigenvalues (λ_i) and eigenvectors ($\vec{v}_i$) of the 2D Hessian or Structure tensor of the input 2D slice (see Eqs. (7) and (8)):

$$S_{\text{in}}(\mathbf{x}) = S(\mathbf{T}(\mathbf{x})) = |\lambda_1 - \lambda_2| \quad (20)$$

$$\alpha_{\text{in}}(\mathbf{x}) = \alpha(\mathbf{T}(\mathbf{x})) = \arccos(\vec{v}_1 \cdot \hat{e}_x) \quad (21)$$

- **Tensor voting**.

17

The output saliency $S_{\text{out}}(\mathbf{x})$ is computed according to the following expression derived from Eqs. (18) and (19):

$$S_{\text{out}}(\mathbf{x}) = S(\mathbf{U}(\mathbf{x})) = \left| \sum_{m=0}^{4} k_m(\alpha_{\text{in}}(\mathbf{x}))(S_{\text{in}}(\mathbf{x}) * V_m(\mathbf{x})) \right| \qquad (22)$$

with $k_m(\alpha_{\text{in}}(\mathbf{x}))$ and $V_m(\mathbf{x})$ given by Eqs. (16) and (17), respectively, and the convolutions being computed in Fourier space.

As mentioned above, our approach to apply TV on a tomogram involves application of this efficient 2D computational procedure to 2D slices of the tomogram along the three major axes. The final output surface saliency for each voxel is taken as the average of the three curve saliency values available. To reduce the distortion caused by the missing wedge, the input saliency values of the voxels with orientation $|\alpha(\mathbf{T}(\mathbf{x}))| > \omega$ are set to zero, with ω being the maximum tilt angle used in the tilt-series acquisition. This constraint is applied for voxels in xz- and yz-planes.

Figure 3 shows the ability of TV to detect membranes correctly even in the presence of membrane-attached structures and to fill the gaps present in membranes when applied to a cellular cryo-tomogram. Supplementary Figure S1 also illustrates that the output saliency proves to be well suited for automated membrane detection.

4. Tensor Voting-based membrane detection

In this section we present the whole procedure devised to detect membranes. It relies on an improved local detector of membranes (or plane-like structures in general) based on TV. The use of TV provides robustness so that at the end of the process membranes are extracted by a simple thresholding operation. This procedure is named TomoSegMemTV and consists of the following stages:

1. *Filtering and scale-space of the input tomogram.*

   This stage aims to reduce noise and filter out structural information smaller than the scale of interest σ. It also helps to ensure that edges and ridges approach Gaussian profiles, which is important for Stage 3. Usually a Gaussian low-pass filtering is used, but more aggressive noise reduction techniques, such as the anisotropic nonlinear diffusion filter (Frangakis and Hegerl, 2001; Fernandez and Li, 2003, 2005) might be used before scale-space so as to substantially remove noise, flatten background and preserve and enhance features of interest. However, to our experience, the actual effect that this filtering has on the performance of the segmentation algorithm is not significant.

2. *Tensor voting on surface saliency.*

   TV is applied to enhance the local surface information provided by the Hessian or the Structure tensor of the scale-spaced tomogram. If the aim is to detect ridges, the Hessian tensor is used as an input to TV. If edges are to be detected, the Structure tensor is used. Note that in the latter case the output surface information will refer to the middle of the edges but it would be otherwise equivalent to the information obtained by the use of the Hessian tensor, as explained in Section 2.2. Hence, the remaining stages proceed in

19

the same way, regardless of the input tensor used. The performance of TV applied to a cryo-tomogram of a synapse is illustrated in Figure 4.

3. *Local surface detection and characterization.*

The NMS detector defined in Eq. (3) is employed to detect membrane structures from the surface saliency obtained by the previous stage. This results in a 1-voxel-thick surface, which represents the centerlines of the potential membranes (Figure 5(E)). Next, the extracted points are subjected to characterization through planarity descriptors, which determine the probability of belonging to a surface. As it was done for vessel detection in MRI (Frangi et al., 1998) and cell membranes in optical microscopy (Mosaliganti et al., 2012), 3D planarity map is defined by the following mathematical expression:

$$P(\mathbf{x}) = \begin{cases} P_s(\mathbf{x}) \cdot P_p(\mathbf{x}), & \forall \mathbf{x} \in \mathcal{S} \\ 0, & \text{otherwise} \end{cases} \tag{23}$$

with

$$P_s(\mathbf{x}) = 1 - e^{-\lambda_1(\mathbf{x})/\hat{\lambda}_1} \quad P_p(\mathbf{x}) = 1 - e^{-p(\mathbf{x})/\hat{p}}, \tag{24}$$

where $\mathbf{x}$ are voxel coordinates, $\mathcal{S}$ is a binary mask resulting from the NMS operation, and $\hat{\cdot}$ is the arithmetic mean of the samples in the $\mathcal{S}$ domain. p is the descriptor that measures planarity of a structure according to the eigenvalues of the Hessian tensor of the scale-spaced volume:

$$p = 1 - \frac{|\lambda_2|}{|\lambda_1|}, \tag{25}$$

P_s enhances points that stand out from the background, whereas P_p those exhibiting locally planar structure. These definitions ensure that the planarity map takes values in the range $[0, 1]$ and that the contribution of factors P_s and P_p to P are equally important. The weight of the factors λ_1

20

and $p(x)$ is automatically controlled through their means in $\mathcal{S}$. The example presented in Figure 5 shows that these descriptors complement each other.

4. *Tensor voting for bridging gaps.*

   It is possible to observe in Figure 5 that the descriptor P gives an output that is similar to the expected segmentation. However, direct thresholding of P may still produce holes in those membrane regions with lower P values. In order to make P more uniform and fill potential gaps, TV is applied on the Hessian tensor of P, using the same parameter configuration as in Stage 2. Prior to TV, P is subjected to Gaussian filtering ($\sigma \approx 0.5$) to make it slightly denser (note that P is sparse as it results from a NMS operation).

5. *Local membrane detector.*

   The map resulting from the previous stage could already be used as a local detector of membranes. However, we find it useful to proceed with the eigen-analysis of the Hessian tensor of that map to extract the absolute value of the largest eigenvalue ($|\lambda_1|$, with $\lambda_1 < 0$; 0 otherwise), followed by NMS. The resulting map (Figure 6(Left)) is more robust than P (Figure 5(D)) in the sense that the difference between the membranes and the other points is higher. This is because $|\lambda_1|$ is more sensitive to the density and can thus detect true membrane points. In addition, the eigen-analysis performed at this stage provides the information about the direction perpendicular to the membrane ($\vec{v_1}$) at all membrane points.

6. *Thresholding and global analysis.*

   This post-processing stage aims to analyze and integrate, at a global scale, the local structural information derived in the previous stages. First, spurious structures have to be rejected. The robustness of the membrane detector

of the previous stage enables this rejection by a simple standard thresholding procedure (Figure 6(Center)). Second, a global analysis, based for instance on volumetric size, helps to discern whether the segmented structures are actual membranes and also to distinguish between different types of membranes, as described previously (Martinez-Sanchez et al., 2011) (Figure 6(Right)). More sophisticated global analysis could be used, such as that based on clustering of the local membrane density profile (along the direction perpendicular to the membrane $\vec{v_1}$) (Martinez-Sanchez et al., 2013).

The output of TomoSegMemTV consists of 1-voxel thick surfaces. The membrane thickness can be easily increased by means of morphological operations.

The whole procedure works in 3D except Stages 2 and 4, which are based on the application of TV to 2D slices, as described earlier. Stages 3 and 5, however, are based on the 3D Hessian tensor (see Section 2) calculated for all voxels of the corresponding input map.

The tuning of the parameters required by the algorithm is relatively simple. The key parameter for TV is σ_v, which is related to the SNR/contrast. The poorer SNR/contrast conditions are, the higher σ_v should be. As this parameter defines the neighbourhood in TV, higher values tend to emphasize larger membranes over smaller ones (for instance, compare synaptic membranes to vesicles in Figure 4). In our experiments, values in the range from 5 to 15 voxels were taken. The threshold t_b on the response of the membrane detector is required at the last stage to produce the output binary map. It can be easily set by the user based on visual inspection (see Figure 6). At this point, a threshold on the volumetric size t_v of the segmented structures may be required to extract the targeted membranes. The scale σ for scale-space operation should be tuned according to the membrane

22

thickness in the tomograms. The parameter ω is given by the maximum tilt angle used in the tilt-series acquisition. With regard to the parameters δ and t_M of the NMS detector (Eq. (3)), in this work they were fixed to 0.71 and 0.03, respectively.

5. Results

5.1. Validation of the method

A quantitative analysis based on a synthetic phantom tomogram under different noise conditions was performed to validate TomoSegMemTV. The evaluation includes a comparison with a standard technique, template matching.

Template matching underlies recent approaches for automated segmentation of specific structures (Fernandez, 2012), including membranes (Lebbink et al., 2007, 2010). In these works, a template suitable for membranes (typically a cuboid shape) is rotated, adjusted to account for the distortion caused by the missing wedge and cross-correlated with the tomogram (Lebbink et al., 2007). The resulting cross-correlation map, which gathers the results from all possible rotated versions of the template, is then thresholded to obtain the final segmentation.

The phantom (ground truth) that we have designed for this validation includes a number of geometric features resembling membranes, vesicles and tubular structures (see Supplementary Figures S3 and S4). To simulate macromolecular crowding, the *Saccharomyces cerevisiae* 80S ribosome structure, taken from the EM databank (http://emdatabank.org; entry emd-1076), was properly scaled, randomly oriented and placed throughout the phantom tomogram. We tested different number of ribosomes (1000, 5000 and 10000) and different noise conditions (SNR=0.05, 0.1 and 2). For template matching, we used the Molmatch implementation (Forster et al., 2010) with a template consisting of a plane with the thickness of the membranes in the phantom. For TomoSegMemTV, σ and σ_v were set to

23

1 and 10, respectively. In order to ensure a fair comparison between template matching and TomoSegMemTV, the same strategy was used for the thresholding and global analysis (see Supplementary Material, Section S.5.1).

The metrics used for the quantitative analysis are the following. First, the fraction of true positives (TPF; also known as sensitivity) is the fraction of membrane points that have been correctly classified. Second, the fraction of true negatives (TNF; also known as specificity) is the fraction of non-membrane points that have been correctly left out of the membranes. Third, the precision (PR) gives the proportion of overlapping/concordance between the phantom and the segmentation (see (Garduno et al., 2008; Martinez-Sanchez et al., 2011, 2013) for more information about these metrics). If H_g and H are the binary representations of the ground truth (i.e. the phantom) and the segmentation result respectively, these metrics are defined as: TPF $= \frac{|H \cap H_g|}{|H_g|}$, TNF $= \frac{|H^C \cap H_g^C|}{|H_g^C|}$, and PR $= \frac{|H \cap H_g|}{|H \cup H_g|}$, where $|\cdot|$ is the number of elements in a set, $\cup$ and $\cap$ represent the union and intersection set operations, respectively, and A^C denotes the complement of set A.

Tables 1 and 2 summarize the results obtained from the evaluation. As expected, the performance of both techniques deteriorates with decreasing SNR and increasing macromolecular crowding. In particular, the latter factor severely degrades the performance of template matching. These results show that TomoSegMemTV is much more robust to noise and crowding and outperforms template matching, having sensitivity (TPF) around 90%. It is remarkable that TomoSegMemTV under the worst conditions (SNR=0.05 and 10000 ribosomes) presents behaviour similar to template matching in the best scenario tested here (SNR=2 and 1000 ribosomes). Supplementary Figures S3 and S4 show examples of the performance of the methods on the phantom.

24

Similar behaviour is observed when applied to experimental cryo-tomograms, as presented in Supplementary Section S.5.1 and Figure S5. There, an example of cryo-tomogram of neuronal synapse shows the limitations of template matching. No optimal threshold is found, leading to false positives and negatives. On the contrary, the thresholding in TomoSegMemTV is straightforward and membranes turn out to be well delineated.

5.2. Applications to experimental datasets

In this section, illustrative examples of the application of TomoSegMemTV to several experimental datasets are shown.

5.2.1. Human Immunodeficiency virus

The segmentation method presented here was applied to a cryo-tomogram of HIV-1 virions that was taken from the EM databank (http://emdatabank.org; entry emd-1155) (Briggs et al., 2006). This tomogram has been often used in the field to test denoising and segmentation algorithms (e.g. (van der Heide et al., 2007; Fernandez, 2009)). The tomogram was first subjected to the membrane detection algorithm that we developed previously, TomoSegMem (Martinez-Sanchez et al., 2011). In general, it allowed good delineation of the outer membranes of the virions, as shown in Figure 7. However, the limitations of the technique are evidenced by some imperfections, such as gaps in the membranes (green arrowheads) or residual structures apposed to them (yellow arrowheads) that arise from the dense material in the interior of the virions.

The use of TomoSegMemTV overcame those limitations. Tensor voting and the planarity characterization make the method more robust and more sensitive to membranes, as shown in Figure 7. This is reflected in the continuity of the membranes of the virions, which are free of gaps and spurious protrusions.

25

5.2.2. *Caulobacter Crescentus*

As an example of detection of a membrane with edge-like profile, we took a tomogram of *Caulobacter Crescentus*, a gram-negative bacterium, from the Cell-Centered database (http://ccdb.ucsd.edu; entry 3647). We focused on the detection of the outer interface of the bacterial wall. To do so, we followed the TomoSeg-MemTV algorithm but, as already described, we used the Structure tensor at Stage 2 of the algorithm instead of the Hessian tensor (see Section 2). The remaining stages of the algorithm proceeded in the same way as if the Hessian tensor was used. Figure 8 shows that TomoSegMemTV succeeded in detecting the outer membrane.

The availability of information about the normal to the membrane for all segmented points opened up the possibility of further analysis. In this particular case, we extracted density profiles in the direction perpendicular to the membrane to obtain an average density profile at a higher resolution. Supplementary Figure S6 shows the average computed from 31400 orientationally aligned profiles that were calculated from the tomogram of *Caulobacter Crescentus,* using TomoSegMemTV. This figure shows peaks corresponding to the inner membrane, the peptidoglycan layer and the outer interface of the bacterial wall.

5.2.3. *Neuronal synapses*

During the course of this work, TomoSegMemTV has been tested on numerous cryo-tomograms of neuronal synapses. Actually, the developments have been motivated in part by the particular characteristics of those datasets (multiple structures attached to the membranes and/or gaps due to the low SNR and contrast, among other factors), which make automated segmentation especially challenging. Such datasets were used to illustrate the methods presented here. Details

26

regarding sample preparation and tomogram generation of these datasets can be found in Supplementary Material, Section S.5.2.3.

Synapses consist of pre- and postsynaptic terminals and contain a number of structural components that have to be segmented from the tomograms. The plasma membranes of the interacting neurons (referred to as pre-synaptic and post-synaptic membranes) were detected by the procedure presented here (Figures 6 and 9). At the synapse, those membranes are directly apposed to each other and are separated by a uniform distance, thus defining the synaptic cleft. Output normals provided by TomoSegMemTV allowed the determination of the shortest distance between two membranes at every point. Synaptic membranes were detected based on their high volumetric size. Their portions that delineate the synaptic cleft were labelled based on the distinctive inter-membrane distance (Figure 9). The presynaptic terminal contains machinery for the neurotransmitter release and comprises vesicles, as well as short filaments that interconnects the vesicles and tether them to the synaptic membrane. The entire presynaptic cytoplasm was segmented using morphological dilation operations, starting from a point selected by the user and using the segmented membranes as boundaries (Figure 9). Because synaptic vesicles are spherical and have uniform size, they were detected by standard template matching procedures, but taking into account the distortion induced by the missing wedge (Bohm et al., 2000). Figure 10 shows 3D views of the segmentations of some datasets.

6. Discussion and Conclusion

Segmentation of electron tomograms recorded from biological samples is important for the interpretation and visualization of their structural components and

27

is necessary for their further computational analysis. Here we present TomoSeg-MemTV, a procedure for membrane segmentation based on Tensor Voting. It relies on an improved local membrane detector that propagates differential information within the neighbourhood of each voxel in an anisotropic way, according to the underlying global structure. This strategy provides the algorithm with the ability to fill gaps present along the membranes. Moreover, it confers robustness against other structures attached to membranes and against the low SNR typically found in cryo-tomograms. The algorithm has been validated and tested on a number of experimental datasets, with special focus on the segmentation of the structures required for the analysis of neuronal synapses.

TomoSegMemTV can detect structures with either ridge or edge profiles, by simply choosing the Hessian tensor or the Structure Tensor based local detector, respectively. Moreover, the algorithm yields not only the set of segmented membrane points, but also the normals to the membranes. This enables more sophisticated structural analyses, such as characterization and classification of membranes according to their local properties (Martinez-Sanchez et al., 2013). TomoSegMemTV can also be used for determination of average membrane profiles, enabling automated tomographic data analysis. Previous approaches only allowed limited averaging, used semi-automated procedures or required sophisticated techniques (Li et al., 2007; Hoffmann et al., 2008; Zuber et al., 2008; Tocheva et al., 2011).

The output yielded by the local detector of TomoSegMemTV turned out to be very robust, which enabled the use of a simple thresholding procedure to select the putative true membrane points. Therefore, this constitutes an advantage over other methods used in the field including our previous detector (Martinez-Sanchez

28

et al., 2011), where a more complex hysteresis thresholding stage was required. Moreover, template matching is a prominent technique that supports segmentation approaches targeting filamentous or membranous structures (Lebbink et al., 2007; Rigort et al., 2012). However, template matching tends to produce spurious correlation peaks in overcrowded environments, particularly under low SNR as is the case in electron cryo-tomography, which limits the performance of the thresholding operation over the cross-correlation map. As a consequence, an optimum threshold cannot be found and the thresholded map often contains residual false positives or gaps in the targeted structures (false negatives). By contrast, the robustness of TomoSegMemTV makes it less susceptible to this problem. An additional advantage over template matching is that methods based on differential geometry are more flexible to adapt to high membrane curvature (Lebbink et al., 2007).

A quantitative evaluation of phantom datasets has been performed using standard quality metrics for the task of membrane detection under realistic noise and crowding conditions. This analysis has confirmed that the algorithm performs well under different noise levels, including very low SNR, and in the presence of a large macromolecular population. Moreover, this assessment has shown that TomoSegMemTV clearly outperforms template matching. These objective results have also been reflected on the application to experimental datasets, where TomoSegMemTV exhibited very good behaviour under different conditions.

An important benefit from the user's point of view concerns parameter setting. Compared to our previous method (Martinez-Sanchez et al., 2011), the number of key parameters is reduced. Some of them are fixed by the experimental imaging conditions, while the tuning of others is straightforward. Essentially, only the

29

following parameters need to be tuned: the scale of Tensor Voting σ_v, which depends on the SNR and the membrane size, the threshold applied to the local membrane detector and the volume threshold used in the global analysis stage, which is determined by the size of membranes.

The implementation of TomoSegMemTV follows a steerable approach to tensor voting, which makes it particularly efficient to deal with the large datasets commonly found in electron tomography. In addition, we have applied multithreading techniques to further accelerate the program (Fernandez, 2008). As a result, tomograms of typical size are segmented in a matter of minutes on standard multicore computers. This is an advantage over template matching, which requires significant processing time (Bohm et al., 2000; Lebbink et al., 2007). A software package implementing TomoSegMemTV has been developed for public use and it is available upon request. The software has been developed in Matlab®, though computational demanding routines were coded in C++ in order to increase the execution speed. A complementary package, SynapSegTools, with Graphical User Interface for intuitive and friendly segmentation of synapsis is also available.

The segmentation procedure presented here may facilitate subsequent higher resolution structural studies. For example, the precise membrane localization and orientation can provide alignment for subtomogram averaging of membrane-associated complexes (e.g. (Pfeffer et al., 2012; Zanetti et al., 2013)). Moreover, our future interests also include further development of the tensor voting algorithm to detect other structures, such as curvilinear components or macromolecular complexes.

30

Acknowledgements

References

Bohm, J., Frangakis, A. S., Hegerl, R., Nickell, S., Typke, D., Baumeister, W., 2000. Toward detecting and identifying macromolecules in a cellular context: Template matching applied to electron tomograms. Proc. Natl. Acad. Sci. USA 97, 14245–14250.

Briggs, J. A., Grunewald, K., Glass, B., Forster, F., Krausslich, H.-G., Fuller, S. D., 2006. The mechanism of HIV-1 core assembly: insights from three-dimensional reconstructions of authentic virions. Structure 14, 15–20.

Cyrklaff, M., Risco, C., Fernandez, J. J., Jimenez, M. V., Esteban, M., Baumeister, W., Carrascosa, J. L., 2005. Cryo-electron tomography of vaccinia virus. Proc. Natl. Acad. Sci. USA 102, 2772–2777.

Fernandez, J. J., 2008. High performance computing in structural determination by electron cryomicroscopy. J. Struct. Biol. 164, 1–6.

Fernandez, J. J., 2009. Tomobflow: feature-preserving noise filtering for electron tomography. BMC Bioinformatics 10:178.

Fernandez, J. J., 2012. Computational methods for electron tomography. Micron 43, 1010–1030.

Fernandez, J. J., Li, S., 2003. An improved algorithm for anisotropic nonlinear diffusion for denoising cryo-tomograms. J. Struct. Biol. 144, 152–161.

Fernandez, J. J., Li, S., 2005. Anisotropic nonlinear filtering of cellular structures in cryoelectron tomography. Comput. Sci. Eng. 7 (5), 54–61.

Forster, F., Han, B. G., Beck, M., 2010. Visual proteomics. Methods Enzymol. 483, 215–243.

Frangakis, A. S., Hegerl, R., 2001. Noise reduction in electron tomographic reconstructions using nonlinear anisotropic diffusion. J. Struct. Biol. 135, 239–250.

Frangi, A. F., Niessen, W. J., Vincken, K. L., Viergever, M. A., 1998. Multiscale vessel enhancement filtering. In: Medical Image Computing and Computer-Assisted Intervention—MICCAI'98. Vol. 1496 of Lecture Notes in Computer Science. Springer, pp. 130–137.

Franken, E., van Almsick, M., Rongen, P., Florack, L., ter Haar Romeny, B., 2006. An efficient method for tensor voting using steerable filters. In: Computer Vision – ECCV 2006. Vol. 3954 of Lecture Notes in Computer Science. Springer, pp. 228–240.

Freeman, W. T., Adelson, E. H., 1991. The design and use of steerable filters. IEEE Trans. Pattern Anal. Mach. Intell. 13, 891–906.

Garduno, E., Wong-Barnum, M., Volkmann, N., Ellisman, M. H., 2008. Segmentation of electron tomographic data sets using fuzzy set theory principles. J. Struct. Biol. 162, 368–379.

Golub, G., van Loan, C., 1996. Matrix Computations. The Johns Hopkins University Press, London, UK.

Hoffmann, C., Leis, A., Niederweis, M., Plitzko, J. M., Engelhardt, H., 2008. Disclosure of the mycobacterial outer membrane: cryo-electron tomography and vitreous sections reveal the lipid bilayer structure. Proc. Natl. Acad. Sci. USA 105, 3963–3967.

33

Lebbink, M. N., Geerts, W. J., van der Krift, T. P., Bouwhuis, M., Hertzberger, L. O., Verkleij, A. J., Koster, A. J., 2007. Template matching as a tool for annotation of tomograms of stained biological structures. J. Struct. Biol. 158, 327–335.

Lebbink, M. N., Jimenez, N., Vocking, K., Hekking, L. H., Verkleij, A. J., Post, J. A., 2010. Spiral coating of the endothelial caveolar membranes as revealed by electron tomography and template matching. Traffic 11, 138–150.

Li, Z., Trimble, M. J., Brun, Y. V., Jensen, G. J., 2007. The structure of FtsZ filaments in vivo suggests a force-generating role in cell division. EMBO J. 26, 4694–4708.

Lucic, V., Rigort, A., Baumeister, W., 2013. Cryo-electron tomography: the challenge of doing structural biology in situ. J. Cell Biol. 202, 407–419.

Martinez-Sanchez, A., Garcia, I., Fernandez, J. J., 2011. A differential structure approach to membrane segmentation in electron tomography. J. Struct. Biol. 175, 372–383.

Martinez-Sanchez, A., Garcia, I., Fernandez, J. J., 2013. A ridge-based framework for segmentation of 3D electron microscopy datasets. J. Struct. Biol. 181, 61–70.

Medioni, G., Lee, M. S., Tang, C. K., 2000. A computational framework for segmentation and grouping. Elsevier Science, Amsterdam, The Netherlands.

Mosaliganti, K. R., Noche, R. R., Xiong, F., Swinburne, I. A., Megason, S. G., 2012. ACME: automated cell morphology extractor for comprehen-

sive reconstruction of cell membranes. PLoS Comput. Biol. 8, e1002780. doi: 10.1371/journal.pcbi.1002780.

Moussavi, F., Heitz, G., Amat, F., Comolli, L. R., Koller, D., Horowitz, M., 2010. 3D segmentation of cell boundaries from whole cell cryogenic electron tomography volumes. J. Struct. Biol. 170, 134–145.

Nurgaliev, D., Gatanov, T., Needleman, D. J., 2010. Automated identification of microtubules in cellular electron tomography. Methods Cell Biol. 97, 475–495.

Pfeffer, S., Brandt, F., Hrabe, T., Lang, S., Eibauer, M., Zimmermann, R., Forster, F., 2012. Structure and 3D arrangement of endoplasmic reticulum membrane-associated ribosomes. Structure 20, 1508–1518.

Rigort, A., Günther, D., Hegerl, R., Baum, D., Weber, B., Prohaska, S., Medalia, O., Baumeister, W., Hege, H.-C., 2012. Automated segmentation of electron tomograms for a quantitative description of actin filament networks. J. Struct. Biol. 177, 135–144.

Sandberg, K., Brega, M., 2007. Segmentation of thin structures in electron micrographs using orientation fields. J. Struct. Biol. 157, 403–415.

Tocheva, E. I., Matson, E. G., Morris, D. M., Moussavi, F., Leadbetter, J. R., Jensen, G. J., 2011. Peptidoglycan remodeling and conversion of an inner membrane into an outer membrane during sporulation. Cell 146, 799–812.

Tong, W. S., Tang, C. K., Mordohai, P., Medioni, G., 2004. First order augmentation to tensor voting for boundary inference and multiscale analysis in 3D. IEEE Trans. Pattern Anal. Mach. Intell. 26, 594–611.

35

van der Heide, P., Xu, X. P., Marsh, B. J., Hanein, D., Volkmann, N., 2007. Efficient automatic noise reduction of electron tomographic reconstructions based on iterative median filtering. J. Struct. Biol. 158, 196–204.

Volkmann, N., 2002. A novel three-dimensional variant of the Watershed transform for segmentation of electron density maps. J. Struct. Biol. 138, 123–129.

Volkmann, N., 2010. Methods for segmentation and interpretation of electron tomographic reconstructions. Methods Enzymol. 483, 31–46.

Weber, B., Greenan, G., Prohaska, S., Baum, D., Hege, H. C., Muller-Reichert, T., Hyman, A. A., Verbavatz, J. M., 2012. Automated tracing of microtubules in electron tomograms of plastic embedded samples of Caenorhabditis elegans embryos. J. Struct. Biol. 178, 129–138.

Weickert, J., 1998. Anisotropic Diffusion in Image Processing. Teubner-Verlag, Stuttgart, Germany.

Zanetti, G., Prinz, S., Daum, S., Meister, A., Schekman, R., Bacia, K., Briggs, J. A. G., 2013. The structure of the COPII transport-vesicle coat assembled on membranes. eLife 2:e00951.

Zuber, B., Chami, M., Houssin, C., Dubochet, J., Griffiths, G., Daffe, M., 2008. Direct visualization of the outer membrane of mycobacteria and corynebacteria in their native state. J. Bacteriol. 190, 5672–5680.

Figure Legends

Figure 1. Response of detectors based on Hessian tensor and Structure tensor. (Left) A ridge profile is shown in blue and the absolute value of the largest eigenvalue of the Hessian Tensor, i.e. second order derivative (the positive values have been set to zero as in Eq. (2)) in red. The dashed line marks the ridge point where the major eigenvalue reaches its local maximum. (Right) A function showing an edge profile in blue and the largest eigenvalue of the Structure Tensor, i.e. first order derivative, in red. The dashed line marks the edge point where the major eigenvalue reaches its local maximum. The values of the eigenvalues have been normalized to fit the range of the function.

Figure 2. Tensor Voting in 2D. (A) Model for vote casting based on a stick tensor. The voter at $\mathbf{x}'$ is shown with its normal in green. The voxel $\mathbf{x}$ is the receiver. The dashed arc represents the arc of a circle passing through $\mathbf{x}'$ and $\mathbf{x}$, which is the most likely smooth path between these points. The orientation of the vote cast from $\mathbf{x}'$ to $\mathbf{x}$ is shown in red. (B) 2D stick voting field calculated for $\sigma_v = 10$. The center of the field is placed at the origin, and the normal oriented along the x-axis. The 8-shape encompasses the votes with most significant saliency. (C) Tensor voting mechanism. Votes cast from three voters (orientation vectors in green) belonging to a curve (black) are shown. The votes are cast to all voxels in the neighbourhood, while the contributions to some receivers belonging to the same curve are shown as red dotted arrows. At the end of the voting process, voxels belonging to the curve (a geometric feature) will have been strengthened, hence enhancing the feature. The other voxels will have received divergent information, which will

37

smear them out.

Figure 3. Performance of tensor voting with cryo-ET data. (Left) a slice of a cryo-tomogram of a synaptic active zone. The green arrow points to a membrane gap. The yellow arrows point to structures anchored to the membranes. (Right) Output of our TV implementation where membranes are enhanced, the gaps are filled and the membrane-attached structures are disregarded. In this example, the parameters were set to $\sigma = 2$ and $\sigma_v = 10$.

Figure 4. Tensor voting applied to an experimental cryo-tomogram of a synapse. (Left) a slice of the original tomogram. (Right) Output of our TV implementation, with σ_v set to 7. Prior to TV, the tomogram was subjected to scale-space operation with $\sigma = 3$. Pixel size at the specimen level = 1.87 nm. Bar = 100 nm.

Figure 5. Local surface characterization. (A) Colormap used for descriptors shown in panels B-E. Blue color corresponds to the minimum value and red to the maximum. (B-D) Planarity descriptors presented in Stage 3 of the TomoSeg-MemTV procedure. (B) P_s of the slice in Figure 4(Left). (C) P_p. (D) P. (E) This panel is obtained by the application of the NMS detector (Eq. (3)) to the output of TV (Figure 4(Right)) and it is used as the input for the computation of planarity descriptors (panels B-D). When compared to (D), the higher robustness of the descriptor P throughout the major membranes becomes apparent.

Figure 6. Final stages of membrane detection with TomoSegMemTV. (Left) Output from the local membrane detector in TomoSegMemTV (Stage 5). Notice that

this output is more robust than *P* shown in Figure 5(D). (Center) Result after thresholding (Stage 6). (Right) Result of the global analysis based on volume (Stage 6): the color encodes the volume measured for every membrane according with the colormap in Figure 5(A).

Figure 7. TomoSegMemTV applied to a cryo-tomogram of HIV-1 virions. (A) Slice of the original tomogram. (B) Scale-space with $\sigma = 2$. (C) Output of the local membrane detector of TomoSegMem (also known as Membrane strength, presented here in Eq. (2)). (D) Final segmentation of TomoSegMem through hysteresis thresholding and global analysis based on size (Martinez-Sanchez et al., 2011). The three virions are shown with different grey values that encode the size of every segmented membrane. Green arrowheads point to gaps and yellow arrowheads point to spurious segmented structures apposed to membranes. (E) Output of the local membrane detector of TomoSegMemTV (Stage 5 in Section 4). (F) Final segmentation of TomoSegMemTV ($\sigma_v = 10$) through simple thresholding and size-based global analysis. The gaps and the structures apposed have disappeared. Grey values represent the size of the segmented membrane. Pixel size at the specimen level = 1.64 nm. Bar = 100 nm.

Figure 8. TomoSegMemTV applied to a tomogram of *Caulobacter Crescentus*. From left to right, the panels show a slice of the original tomogram, the delineated membrane and a 3D view of the segmented structure ($\sigma_v = 10$ and $\sigma = 3$). Pixel size at the specimen level = 2.84 nm. Bar = 400 nm.

Figure 9. Segmentation by TomoSegMemTV of three different cryo-tomograms of neuronal synapses. We used $\sigma_v \in [5, 10]$ and $\sigma \in [1.5, 2]$. These datasets were denoised by anisotropic nonlinear diffusion prior to the segmentation. (A,B) Tomogram 1, shown already on Figure 6. (A) Detection of the pre- and post-synaptic membranes (green) and the membrane regions that delineate the synaptic cleft (red). Note the little effect provided by the prior use of anisotropic nonlinear diffusion when compared to Figures 4 and 6. (B) Pre-synaptic cytoplasm (blue) was segmented using dilation operations from a point selected by the user (see panel A) and with the segmented membranes acting as boundaries. Part of the cytoplasm that lies within a specified distance to the synaptic cleft is highlighted in cyan. The vesicles (yellow) were detected by template matching using spheres distorted by the missing wedge as references. (C-F) Tomogram 2. (C) Slice of the tomogram. The arrow points to an area zoomed in (E,F). (D) Segmentation. The same colors as in (A,B) are used to denote the segmented membranes and regions. In violet, a mitochondrion is extracted with TomoSegMemTV configured for detecting edge-like structures. (E) Membranes in tomograms often present holes or appear blurred. Tensor Voting succeeded in bridging these gaps (F). (G,H) Tomogram 3. Segmented structures use the same color code as above. The panels also show the graphical user interface developed in this work. Pixel size at the specimen level = 1.87 nm.

Figure 10. 3D visualization of several segmented cryo-tomograms of neuronal synapses. Segmented membranes are shown in blue, the boundaries of the synaptic cleft in red and vesicles in yellow.

40

Figures

Figure 1: Response of detectors based on Hessian tensor and Structure tensor. (Left) A ridge profile is shown in blue and the absolute value of the largest eigenvalue of the Hessian Tensor, i.e. second order derivative (the positive values have been set to zero as in Eq. (2)) in red. The dashed line marks the ridge point where the major eigenvalue reaches its local maximum. (Right) A function showing an edge profile in blue and the largest eigenvalue of the Structure Tensor, i.e. first order derivative, in red. The dashed line marks the edge point where the major eigenvalue reaches its local maximum. The values of the eigenvalues have been normalized to fit the range of the function.

Figure 2: Tensor Voting in 2D. (A) Model for vote casting based on a stick tensor. The voter at $\mathbf{x}'$ is shown with its normal in green. The voxel $\mathbf{x}$ is the receiver. The dashed arc represents the arc of a circle passing through $\mathbf{x}'$ and $\mathbf{x}$, which is the most likely smooth path between these points. The orientation of the vote cast from $\mathbf{x}'$ to $\mathbf{x}$ is shown in red. (B) 2D stick voting field calculated for $\sigma_v = 10$. The center of the field is placed at the origin, and the normal oriented along the x-axis. The 8-shape encompasses the votes with most significant saliency. (C) Tensor voting mechanism. Votes cast from three voters (orientation vectors in green) belonging to a curve (black) are shown. The votes are cast to all voxels in the neighbourhood, while the contributions to some receivers belonging to the same curve are shown as red dotted arrows. At the end of the voting process, voxels belonging to the curve (a geometric feature) will have been strengthened, hence enhancing the feature. The other voxels will have received divergent information, which will smear them out.

Figure 3: Performance of tensor voting with cryo-ET data. (Left) a slice of a cryo-tomogram of a synaptic active zone. The green arrow points to a membrane gap. The yellow arrows point to structures anchored to the membranes. (Right) Output of our TV implementation where membranes are enhanced, the gaps are filled and the membrane-attached structures are disregarded. In this example, the parameters were set to $\sigma = 2$ and $\sigma_v = 10$.

Figure 4: Tensor voting applied to an experimental cryo-tomogram of a synapse. (Left) a slice of the original tomogram. (Right) Output of our TV implementation, with σ_v set to 7. Prior to TV, the tomogram was subjected to scale-space operation with σ=3. Pixel size at the specimen level = 1.87 nm. Bar = 100 nm.

Figure 5: Local surface characterization. (A) Colormap used for descriptors shown in panels B-E. Blue color corresponds to the minimum value and red to the maximum. (B-D) Planarity descriptors presented in Stage 3 of the TomoSegMemTV procedure. (B) P_s of the slice in Figure 4(Left). (C) P_p. (D) P. (E) This panel is obtained by the application of the NMS detector (Eq. (3)) to the output of TV (Figure 4(Right)) and it is used as the input for the computation of planarity descriptors (panels B-D). When compared to (D), the higher robustness of the descriptor P throughout the major membranes becomes apparent.

Figure 6: Final stages of membrane detection with TomoSegMemTV. (Left) Output from the local membrane detector in TomoSegMemTV (Stage 5). Notice that this output is more robust than P shown in Figure 5(D). (Center) Result after thresholding (Stage 6). (Right) Result of the global analysis based on volume (Stage 6): the color encodes the volume measured for every membrane according with the colormap in Figure 5(A).

Figure 7: TomoSegMemTV applied to a cryo-tomogram of HIV-1 virions. (A) Slice of the original tomogram. (B) Scale-space with $\sigma = 2$. (C) Output of the local membrane detector of TomoSeg-Mem (also known as Membrane strength, presented here in Eq. (2)). (D) Final segmentation of To-moSegMem through hysteresis thresholding and global analysis based on size (Martinez-Sanchez et al., 2011). The three virions are shown with different grey values that encode the size of every segmented membrane. Green arrowheads point to gaps and yellow arrowheads point to spurious segmented structures apposed to membranes. (E) Output of the local membrane detector of To-moSegMemTV (Stage 5 in Section 4). (F) Final segmentation of TomoSegMemTV ($\sigma_v = 10$) through simple thresholding and size-based global analysis. The gaps and the structures apposed have disappeared. Grey values represent the size of the segmented membrane. Pixel size at the specimen level = 1.64 nm. Bar = 100 nm.
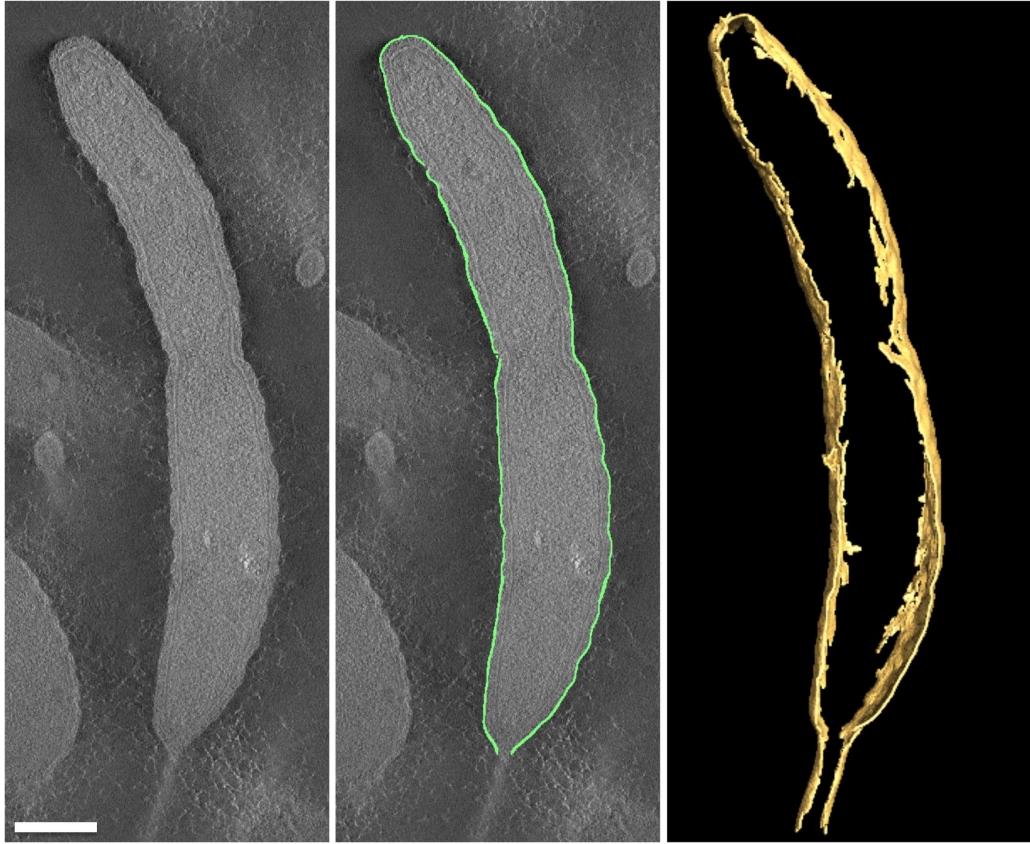
48

Figure 8: TomoSegMemTV applied to a tomogram of *Caulobacter Crescentus*. From left to right, the panels show a slice of the original tomogram, the delineated membrane and a 3D view of the segmented structure (σ_v = 10 and σ = 3). Pixel size at the specimen level = 2.84 nm. Bar = 400 nm.
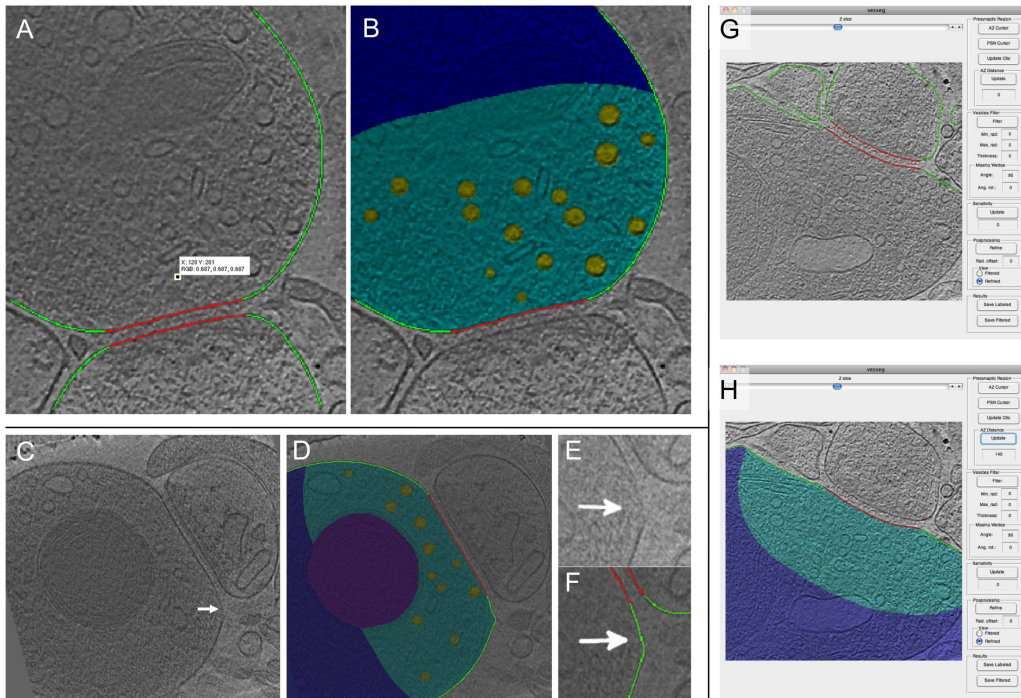
Figure 9: Segmentation by TomoSegMemTV of three different cryo-tomograms of neuronal synapses. We used $\sigma_v \in [5, 10]$ and $\sigma \in [1.5, 2]$. These datasets were denoised by anisotropic nonlinear diffusion prior to the segmentation. (A,B) Tomogram 1, shown already on Figure 6. (A) Detection of the pre- and post-synaptic membranes (green) and the membrane regions that delineate the synaptic cleft (red). Note the little effect provided by the prior use of anisotropic nonlinear diffusion when compared to Figures 4 and 6. (B) Pre-synaptic cytoplasm (blue) was segmented using dilation operations from a point selected by the user (see panel A) and with the segmented membranes acting as boundaries. Part of the cytoplasm that lies within a specified distance to the synaptic cleft is highlighted in cyan. The vesicles (yellow) were detected by template matching using spheres distorted by the missing wedge as references. (C-F) Tomogram 2. (C) Slice of the tomogram. The arrow points to an area zoomed in (E,F). (D) Segmentation. The same colors as in (A,B) are used to denote the segmented membranes and regions. In violet, a mitochondrion is extracted with TomoSegMemTV configured for detecting edge-like structures. (E) Membranes in tomograms often present holes or appear blurred. Tensor Voting succeeded in bridging these gaps (F). (G,H) Tomogram 3. Segmented structures use the same color code as above. The panels also show the graphical user interface developed in this work. Pixel size at the specimen level = 1.87 nm.
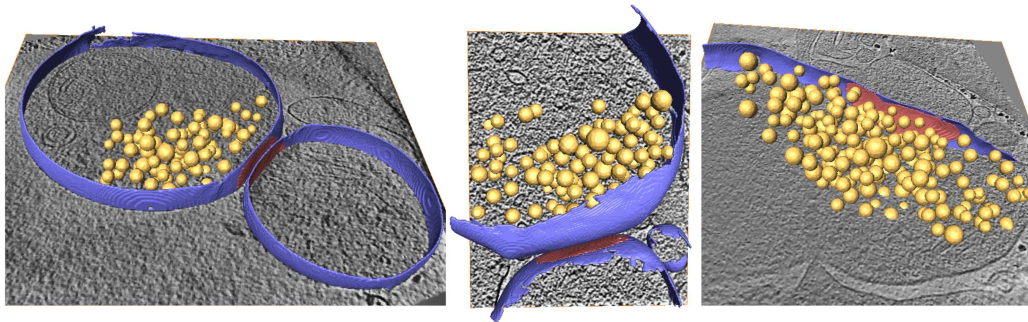
50

Figure 10: 3D visualization of several segmented cryo-tomograms of neuronal synapses. Segmented membranes are shown in blue, the boundaries of the synaptic cleft in red and vesicles in yellow.

Tables

Table 1: **Quantitative analysis based on synthetic phantom for TomoSegMemTV.**

SNR	1000 Ribosomes			5000 Ribosomes			10000 Ribosomes		
	PR	TPF	TNF	PR	TPF	TNF	PR	TPF	TNF
0.05	0.8434	0.9167	0.9947	0.8266	0.9096	0.9941	0.7193	0.8306	0.9906
0.1	0.8474	0.9231	0.9946	0.8332	0.9116	0.9938	0.8004	0.8915	0.9925
2	0.8487	0.9274	0.9939	0.8487	0.9274	0.9939	0.8373	0.9178	0.9937

Table 2: **Quantitative analysis based on synthetic phantom for Template Matching.**

	1000 Ribosomes			5000 Ribosomes			10000 Ribosomes		
SNR	PR	TPF	TNF	PR	TPF	TNF	PR	TPF	TNF
0.05	0.7538	0.8524	0.9921	0.6440	0.7639	0.9878	0.3953	0.4753	0.9867
0.1	0.7640	0.8769	0.9903	0.6859	0.8209	0.9871	0.4671	0.5300	0.9911
2	0.7766	0.8655	0.9932	0.6920	0.8158	0.9882	0.5460	0.6324	0.9896