ORIGINAL PAPER

A tool for the morphological analysis of mixtures of lipids and water in computer simulations

Marc Fuhrmans · Siewert-Jan Marrink

Abstract When analyzing computer simulations of mixtures of lipids and water, the questions to be answered are often of a morphological nature. They can deal with global properties, like the kind of phase that is adopted or the presence or absence of certain key features like a pore or stalk, or with local properties, like the local curvature present at a particular part of the lipid/water interface. While in principle all of the information relating to the global and local morphological properties of a system can be obtained from the set of atomic coordinates generated by a computer simulation, the extraction of this information is a tedious task that usually involves using a visualization program and performing the analysis by eye. Here we present a tool that employs the technique of morphological image analysis (MIA) to automatically extract the global morphology—as given by Minkowski functionals—from a set of atomic coordinates, and creates an image of the system onto which the local curvatures are mapped as a color code.

Keywords Minkowski functionals · Phase determination · Molecular dynamics · Image analysis · Local curvature

M. Fuhrmans · S.-J. Marrink (✉)
Groningen Biomolecular Sciences and Biotechnology
Institute & Zernike Institute for Advanced Materials,
University of Groningen,
Nijenborgh 4,
9747 AG Groningen, The Netherlands
e-mail: s.j.marrink@rug.nl

M. Fuhrmans
e-mail: fuhrmans@rug.nl

Introduction

Motivation

With the development of new models and the steady increase in available computing power, computer simulations have become more and more valuable in the study of lipid systems. While the exact conformations of individual lipid molecules are of interest for some applications, most of the time the focus is on the behavior of aggregates of lipids as a whole. Recent examples have been reviewed in [1].

In many of these studies, at some point during the analysis of the simulation, a morphological property of the system—i.e., a property that solely depends on the shape of the lipid aggregate—needs to be characterized. For the more general properties, like the phase adopted and the presence or absence of stalks or pores, the task at hand can be accomplished by loading the obtained coordinates into a visualization program and performing the analysis by eye, but analyzing a large number of simulations in this way can be a tedious task. For the determination of more specific, quantitative properties like the interface area, volume and curvatures, such a naive approach is largely impossible.

One possible way to automate morphological analyses of trajectories generated by computer simulations is to use the technique of morphological image analysis [2] to extract morphological information in the form of Minkowski functionals [3]. This approach has been used to study, e.g., a pore distribution [4] and membrane fusion events during a phase transition [5], as well as to monitor the self-assembly of vesicles [6]. Another approach is to describe morphological features as persistent voids based on the theory of alpha shapes [7] and persistent homology [8], which has been applied to characterize vesicle fusion [9].

However, no implementation of either method is currently readily available to the majority of researchers—none are included in any of the widely used molecular dynamics software packages.

Here, we present an extension of the Gromacs software package [10] that enables the morphological image analysis of molecular aggregates. In addition, an option to extract local curvatures has been added to the method which, to the authors' best knowledge, has not been employed before, at least in the field of lipid aggregates.

Theory

In three dimensions, there are four Minkowski functionals corresponding to the volume whose morphology is to be determined, the area of the interface separating that volume from the rest of the system, and the integrated mean and Gaussian curvatures of that interface.[1] As such, both geometrical (shape) and topological features (connectivity) are characterized.[2]

For black and white digital (i.e., pixelated) images, the process used to extract the Minkowski functionals is well established and can be accomplished by simply counting the pixels and pixel components of lower dimensionality that comprise the image. This means, that for three-dimensional pictures, one only needs the number of voxels[3] and the number of faces, edges and vertices which these voxels consist of, where voxel components shared by several voxels are counted only once. The Minkowski functionals can then be obtained as sums over these numbers, as given in Table 1. A way of obtaining the morphology of a set of coordinates is therefore to translate the system into a three-dimensional image composed of black and white voxels [2].

The advantages of this method are the straightforwardness of its implementation and its rigorousness in the sense that the resulting numbers are the exact values of the Minkowski functionals for the image. Its only disadvantage is therefore the approximation introduced by the image itself. The use of voxels entails a limitation to right angles, which imposes restrictions on the values for the surface area and integrated mean curvature obtained with this method, causing several structures to share the same value. As an example, removing any voxel from a cube of eight voxels will leave the surface area and integrated mean

Table 1 The relation between volume V, surface area A, mean breadth B, Euler characteristic χ, integrated mean curvature H, integrated Gaussian curvature K, voxel edge length ξ, and the numbers of cubic voxels n_c, faces n_f, edges n_e and vertices n_v that define the positive space

Morphological property	Related property
$V/\xi^3 = n_c$	
$A/\xi^2 = -6n_c + 2n_f$	
$2B/\xi = 3n_c - 2n_f + n_e$	$H = 2\pi B$
$\chi = -n_c + n_f - n_e + n_v$	$K = 4\pi\chi$

curvature unchanged, resulting in a general tendency to overestimate these functionals.

However, the Euler characteristic—which only requires the connectivity to be identical for the image and the original system—can be determined exactly, and the volume can be obtained with only slight errors that can be minimized by choosing a sufficiently high resolution.

For a broad spectrum of morphological tasks, the values obtained are sufficient, even with the restrictions mentioned above. For most applications concerning molecular aggregates, the Euler characteristic and the integrated mean curvature are arguably the most important values. Purely topological analyses, including both phase determination and the detection of stalks or pores, rely primarily on the Euler characteristic, which is not affected by the limitations of morphological image analysis. In addition, due to the systematic nature of the error in the integrated mean curvature, the value obtained can still be used to extract morphological information. The absence of mean curvature is accurately recognized as zero mean curvature, and systems with positive can be distinguished from those with negative total mean curvature. In addition, both the integrated mean curvature and the surface area can be used to further characterize structures within families with similar topologies, since the lack of absolute values is not detrimental to relative comparisons.

As an extension to this basic application of morphological image analysis, it is also possible to obtain local values of the mean and Gaussian curvature. As has been shown by Hyde et al. [11], every surface vertex can be associated with a certain mean and Gaussian curvature. Again, these values are exact for the image, and summation over all surface vertices while taking into account the different surface areas associated with each vertex leads to global (integrated) values for the mean and Gaussian curvatures which are identical to those obtained with the method described above. Mapping the local curvatures onto the image as a color code allows further characterization of the structure at hand, enabling the easy detection of areas with different curvatures, as well as detailed comparison of similar structures.

---

[1] Alternatively, integrated mean and Gaussian curvatures can be replaced by the mean breadth and the Euler characteristic, to which they are proportional (see Table 1).

[2] Note, however, that while a given morphology specifies a specific set of Minkowski functionals, the reverse is not necessarily true.

[3] In other words, the three-dimensional analog of pixels, essentially small cubes.

The rest of this article is organized as follows. In the sections "Implementation" and "User-definable options and parameters," details about the implementation and the user-definable parameters are given, while "Simulation setup" describes the parameters used in the simulations that were analyzed in order to test our program. The "Results" section provides the results of these sample applications, in addition to results of tests performed on model systems.

Methods

Implementation

The implementation discussed in this publication was realized using the Gromacs 3.3 software package [10], but should in principle compile with any version of Gromacs from 3.0 to date, with only minor modifications. The executable is called *g_mia* and was written in the C programming language. The source code is available upon request. Acceptable input file formats are the standard formats supported by Gromacs.

Basic algorithm

We treat the image as a three-dimensional cubic grid representing the simulation box, onto which every coordinate is mapped.[4] To avoid any artificial empty spaces caused by representing atoms (or groups of atoms in the case of coarse-grained models) by their centers of mass only, every coordinate is expanded into a spherical cloud of coordinates, each of which is mapped onto the grid individually.[5] Depending on the type and number of particles mapped to it, cells are declared to be either positive or negative, where positive cells represent the molecular aggregate. The global values of the Minkowski functionals can then be obtained by counting the number of cubes, cube faces, edges and vertices, taking into account the periodic boundaries.

For the local values of the mean curvature and Gaussian curvature, every surface vertex[6] is identified as being one of the possible cases listed in Fig. 1, and the corresponding

local curvatures given by the product of the interface area and the curvature value associated with that type of surface vertex are stored. However, we wish to map the curvature to voxels, not vertices. To that end, nonsurface voxels (i.e.,
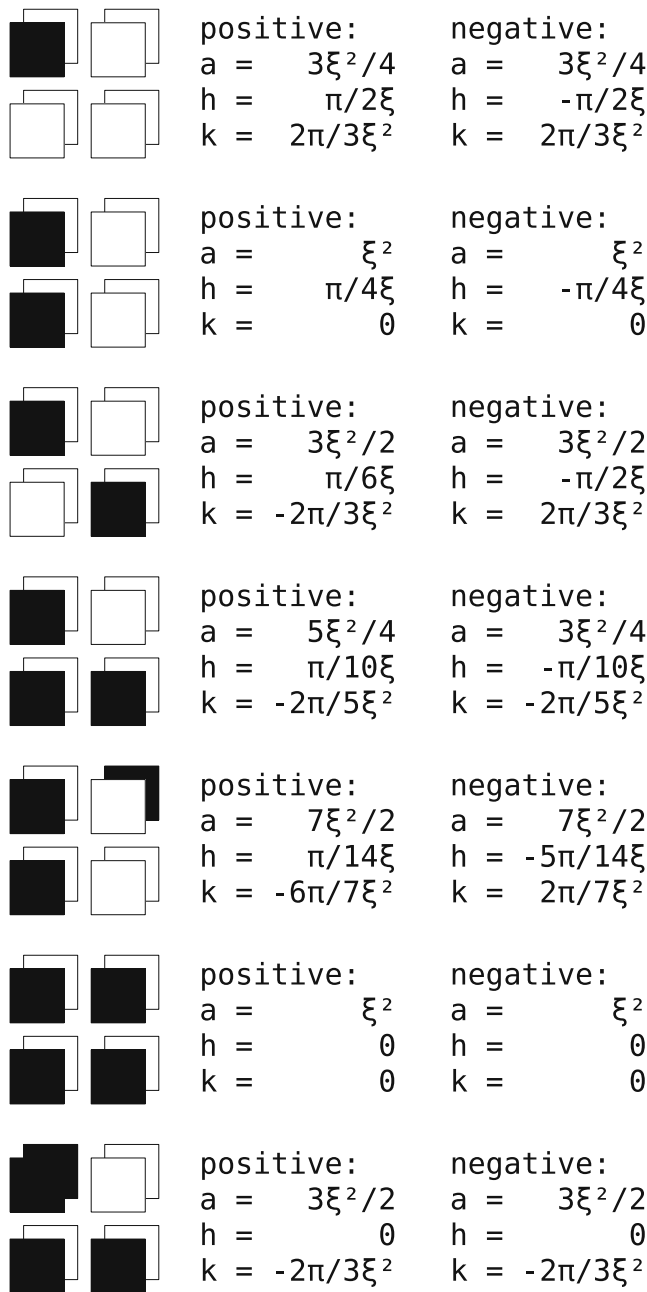

Fig. 1 Overview of the possible types of surface vertices and the associated local values of surface area *a*, mean curvature *h* and Gaussian curvature *k* in relation to the edge-length ξ (adapted from [11]). For each pattern, values are given both for the positive (*black* represents lipids) and the negative case (*black* represents water). While more patterns are possible in principle, these represent noise and should not occur as long as the resolution used does not exceed the coordinate density

---

[4] Assuming that the chosen grid resolution is sufficiently high, the distortion introduced by a potential mismatch between the grid and the simulation box and by ensuring that the periodic boundary conditions for nonrectangular boxes can be mapped is only minor.

[5] In the current implementation, the cloud is generated via a simple loop over spherical coordinates, generating N^3 coordinates within N equidistant shells containing N^2 coordinates each, corresponding to a density distribution that decays towards the perimeter.

[6] In other words, every vertex that belongs to both negative and positive voxels.

positive voxels that do not contribute a single face to the interface) are eliminated. The stored curvatures of the surface vertices are then distributed equally among the surface voxels adjacent to that particular vertex, as illustrated in Fig. 2.

To visualize the local curvatures, a PyMOL [12] file is generated that represents the image as voxels onto which the curvatures are mapped as a color code. Due to the different ranges of curvatures encountered, it is impossible to use a fixed color scale. We therefore employ a two-color scheme in which white corresponds to a curvature of zero while the two colors are used to distinguish negative and positive curvatures, with the intensity of the color indicating the value. Full intensity is assigned to the voxel(s) with the maximum absolute curvature encountered in a given system, and the color range is symmetric in the sense that full intensity indicates the same (absolute) value for both colors. While this means that every image has its own color code, it is the most efficient scheme to highlight differences in local curvature.

Optional steps

The data generated can often be improved considerably by performing some image manipulation steps and averaging.

Image manipulation Depending on the particle density in the coordinate file and the desired resolution of the grid, it is possible to include an image manipulation step right after the creation of the image. In this step, isolated
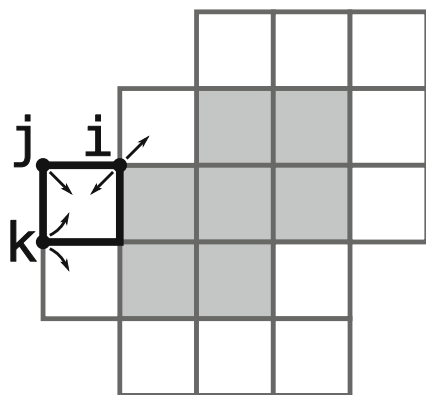
clusters of either positive or negative cells below a certain size are interpreted as noise and removed. Performing this step also allows the number of actual isolated clusters above the threshold size to be determined at no additional cost, which is useful morphological information in its own right.

Spatial averaging Due to the fixed nature of the grid, even aggregates with perfectly homogeneous curvature, like a sphere, will display different curvatures for different regions, depending on how well the rasterization of the image fits the surface in that region. In general, the curvature tends to be underestimated when the surface is aligned with the grid, and overestimated when it is diagonal to the grid.

Two spatial averaging options can be employed to reduce this effect. First, the local curvature obtained can be averaged over neighboring surface voxels within a certain distance. In addition, it is possible to further improve the results by determining local curvatures for multiple grid orientations. In this case, the resulting curvature values of each positive surface voxel for every orientation are stored together with the coordinate corresponding to the center of that voxel rotated back to the original orientation. The values of all rotations are then mapped back onto the original grid, averaging the values over the entries mapped onto the same cell. If needed, the resulting values can be averaged over neighboring cells. Since it is not possible to preserve the periodic boundary conditions with a rotated grid, the area of interest is centered in the box, and only cells within a certain distance from the center (i.e., cells that lie within both the volume of the box and the rotated grid for all rotations) are taken into account.

Time averaging While not included as such in the current version of the presented tool, it can also be useful to average the curvatures over time (i.e., over several snapshots of a trajectory). For the global values, this can easily be accomplished after analysis by taking the floating average of the calculated curvatures. For the local values, time averaging can be performed at the coordinate level prior to the analysis, effectively yielding time-averaged curvatures.

User-definable options and parameters

It is not generally possible to use the same set of parameters for the analysis of all possible structures and representations. The implementation therefore allows most parameters to be determined by the user. This section describes the parameters and discusses what to consider to achieve the optimal results. The corresponding command line options are given in parentheses.

Fig. 2 Mapping of local curvature from surface vertices to voxels. After eliminating nonsurface voxels (*gray-shaded squares*), the curvature that was calculated for the surface vertices is distributed equally among all adjacent surface voxels (*arrows*). The local curvature C of the highlighted voxel is obtained as $C = (1/2)i + j + (1/2)k$, where i, j and k are the curvatures corresponding to the surface vertices adjacent to the highlighted voxel. A two-dimensional example is shown for the sake of simplicity, but the method also applies to three dimensions

Input files The tool needs a coordinate or trajectory file (-c) and an index file (-n) in which the particles that correspond to the positive phase are listed.

Imaging options The edge length of the grid (-dim), the radius of the spherical cloud used to expand the coordinates (-sr), and the number of coordinates generated during the expansion (-npts), as well as the minimum number of coordinates mapped onto a grid cell required to count it as positive (-thresh1) need to be specified.

As a general consideration, the resolution needs to be high enough to accurately depict the structure to be analyzed, but is limited by memory requirements, due to the need for several three-dimensional arrays during the computation.[7] In addition, using a high resolution usually requires the expansion of the coordinates in order to avoid the creation of artificial empty voxels due to the limited coordinate density, which partially offsets the desired high resolution. The radius of the spherical cloud should therefore be chosen as the smallest radius sufficient to avoid noise. (An example of the effects of the chosen resolution for a sample application is given in "Applications," Table 2)

It also turns out that, in order to accurately detect flat morphologies with zero mean curvature, it is necessary to calibrate the parameters used. Since molecular aggregates usually have low short-range order, fluctuations of individual molecules from the mean will show up as either bumps or dents in the created image. Since a given resolution does not necessarily have the same propensity to produce bumps as it does to produce dents, a net curvature will be measured. The threshold parameter can be used to adjust the number of "positive" coordinates that must be mapped onto a single grid cell to count that cell as positive in order to (on average) produce an equal number of bumps and dents, thus ensuring that an artificial mean curvature is not introduced into the measurement.

In addition, it is also possible to use the coordinates of the particles corresponding to the negative phase—mapping them onto the grid as described above, but counting them as negative instead. If that is desired, the number of phases to consider must be set from 1 to 2 (-np), and the index file needs to contain a second group in which these particles are listed.

If isolated clusters below a certain size are to be removed (see above), the maximal cluster size that is considered noise must be specified (-cs).

Averaging options The range over which the local curvatures are averaged over neighboring voxels needs to be specified (-ar1 and -ar2), with a value of zero indicating no averaging. Two values are needed, one for the averaging of every single grid orientation (-ar1) and one for the averaging performed after the values of all grid orientations have been collected (-ar2).

If multiple grid orientations are to be used, the number of rotations around every axis (-nx, -ny and -nz) and the corresponding angle increments (-depsilon, -dphi and -dtheta), as well as the radius around the center of the box within which the voxels are considered must be set (-dr).[8] In order to achieve the best result, care must be taken to avoid sampling similar orientations.

In addition, it is possible to specify a threshold which ensures that voxels are only counted as positive if a minimum number of local curvatures corresponding to different rotations have been mapped onto that voxel (-thresh2). However, unlike the other averaging steps, this option will discard curvature and does not yield exact results, and should therefore be used with care. For the results presented in this work, a threshold of zero has been used, effectively disabling this option.

For the results discussed in the "Results" section, the grid resolution and the radius used to expand the coordinates will be given, along with the number of rotations and the distance used to average the local values.

Simulation setup

The simulations shown in this article were performed using the coarse-grained MARTINI model [13] with the Gromacs 3.3 software package [10], employing the standard run parameters for the MARTINI model at a timestep of 40 fs. Both pressure and temperature were coupled to a reference value using the Berendsen scheme [14]. Lennard-Jones and Coulomb interactions were obtained at every step for particles occurring within a cut-off of 1.2 nm according to a neighbor list that was updated every 10 steps. The Lennard-Jones and the Coulomb potentials were modified with a shift function to ensure that the interactions vanished smoothly at the cut-off. Electrostatic interactions were screened with an effective dielectric constant of 15 (which is the standard value for the MARTINI model).

Three processes were used as sample applications: spontaneous aggregation of lipids into a lipid bilayer, closure of a pore in a membrane, and stalk formation between apposed lipid bilayers (with setups similar to those used for the simulations described in [15–17], respectively).

---

[7] No attempt was made to optimize the code in this respect.

[8] This radius needs to be specified as a value between 0 and 1, and it will be multiplied by half of the smallest box dimension internally.

Spontaneous aggregation The system simulated consists of 256 DOPE (dioleoylphosphatidylethanolamine) lipids with 768 water beads (with one bead corresponding to four water molecules), starting from random coordinates. The simulation was carried out at a reference temperature of 315 K, with a coupling time constant of 0.5 ps, anisotropic pressure coupling, compressibilities of 5×10^{-5} bar^{-1} for the diagonal elements and 1×10^{-7} bar^{-1} for the off-diagonal elements of the pressure tensor, coupling time constants of 1.2 ps, and reference pressures of 1.0 bar.

Porated membrane The system consists of a bilayer of 128 DPPC (dipalmitoylphosphatidylcholine) lipids with a pre-formed pore at excess hydration (2653 water beads). After a short equilibration, the simulation was carried out at a reference temperature of 323 K with a coupling time constant of 1.0 ps, semi-isotropic pressure coupling with a compressibility of 1×10^{-5} bar^{-1}, a coupling time constant of 1.0 ps, a reference pressure of 1.0 bar for the direction perpendicular to the bilayer, and a compressibility of 0 bar^{-1} for the plane containing the bilayer.

Stalk formation The initial configuration was two bilayers of 98 DOPE lipids each, separated by two slabs consisting of 65 water beads each, corresponding to an effective hydration level of 2.65 water molecules per lipid. To induce the formation of stalks, the simulation was carried out at a reference temperature of 375 K with a coupling time constant of 0.5 ps, semi-isotropic pressure coupling with a compressibility of 1×10^{-5} bar^{-1}, a coupling time constant of 1.2 ps, and a reference pressure of 1.0 bar for all directions.

Results

Model systems

The method was first tested on two artificially constructed model systems with very high coordinate densities: a solid sphere and a toroidal pore. This allowed the potential of the method to be assessed by analyzing virtually noise-free structures, and meant that the exact values for these ideal geometries were available for comparison. Plots of the coordinates of the model systems used are depicted in Figs. 3 and 4.

Spheres Figure 5 shows the measured and theoretical values of the Minkowski functionals for solid spheres of different radii. The image was constructed with a resolution of 0.4 nm and by expanding the coordinates into spheres of 0.2 nm. As predicted for a solid object, the
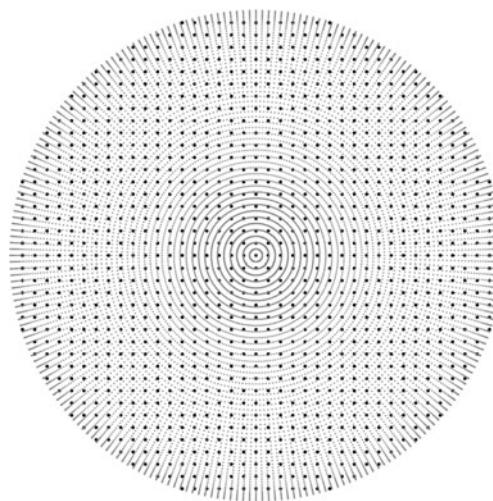

Fig. 3 The spherical model system. The volume of the sphere was filled with coordinates at a regular distance ($d=r/20$) that depends on the radius r (*large black dots*). In addition, the surface (the most important part) was covered with coordinates at a very high density (*small gray dots*)

Euler characteristic is obtained with a value of exactly 1. The volume of the image is only slightly higher than that of the original, which is due to the rasterization of the image and the expansion of the coordinates into spheres. However, the surface area and integrated mean curvature are overestimated to a larger extent. In fact, the values obtained lie between the values of the sphere and a cube with an edge length identical to the diameter of the sphere (see discussion in "Theory"). Nevertheless, the values are proportional to the values of the original and could therefore be used in principle to distinguish between spheres with different sizes.

To calculate the local curvatures, eight rotations around every axis were used, and the values were averaged over neighboring voxels up to a distance of three grid cells. Looking at the mapping onto the image shown in Fig. 6, we can see that both the mean and the Gaussian curvatures are accurately mapped with positive values. While the mean curvature is correctly mapped almost homogeneously over the whole surface, the distribution of the Gaussian curvature for the larger sphere is less even, even with averaging performed. This is a symptom of a general difficulty with mapping the Gaussian curvature that was found in most of our measurements for systems which display large areas of homogeneous Gaussian curvature.[9] However, while this behavior might seem problematic at

---

[9] This is also true, to a lesser degree, for the mean curvature; however, the values of the mean curvature tend to be higher and therefore less sensitive to artificial fluctuations caused by the rasterization.
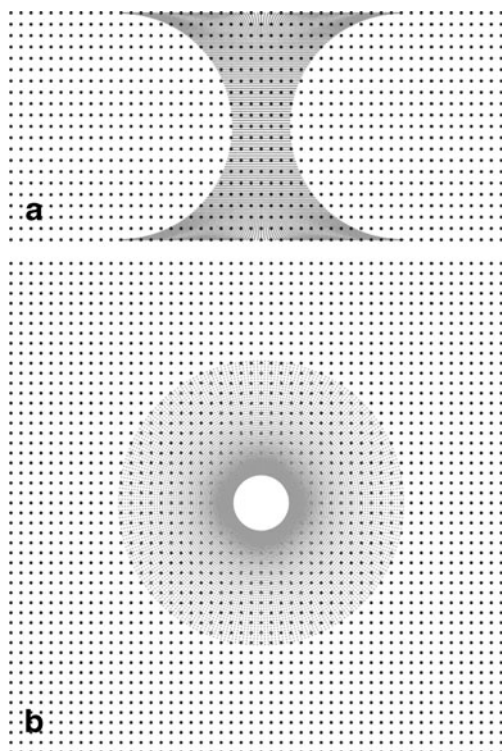
Fig. 4a–b The model system for the ideal toroidal pore (a *xz* plane, b *xy* plane). The volume of the porated slab was filled with coordinates at a regular distance of 0.176 nm in the *x* and *y* directions and 0.2 nm in the *z* direction (*large black dots*). In addition, the curved surface (the most important part) was covered with coordinates at a very high density (*small gray dots*)

first, it is partially due to the color scale employed, which assigns full color intensity to the voxel with the highest absolute curvature (see "Implementation"). In the presence of regions with high Gaussian curvature (as in the example
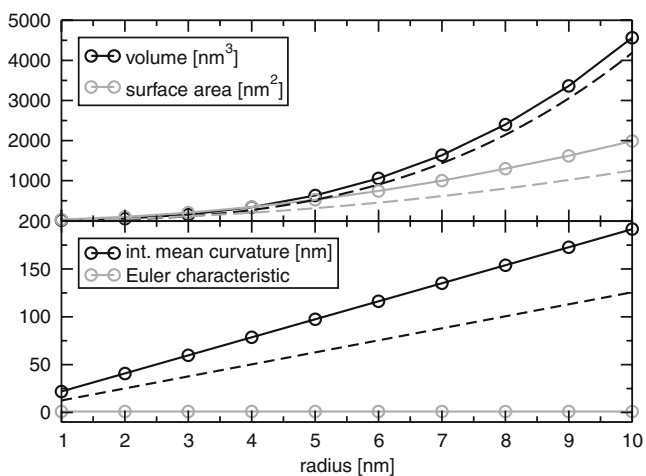
Fig. 5 Morphological image analysis of model systems of solid spheres of different radii (*lines between measured points* serve to guide the eye). Calculated values corresponding to the underlying geometry are plotted (*dashed lines*) for comparison
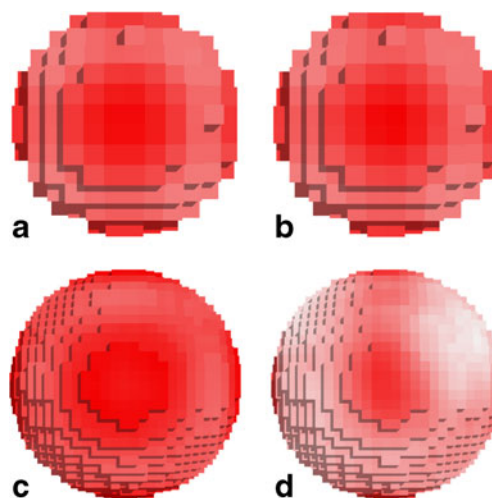

Fig. 6a–d Mapping the local values of mean (**a**, **c**) and Gaussian (**b**, **d**) curvature onto the constructed images of spherical model systems. Shown are the results for spheres with radii of 2 nm (**a**, **b**) and 5 nm (**c**, **d**). Positive curvature is depicted in *red*, zero curvature in *white*, and negative curvature in *blue*

of the smaller sphere), these are accurately detected, and artificial fluctuations in regions of lower Gaussian curvature become relatively less important as well as less visible in our depiction.

Toroidal pores Figure 7 shows the values of the Minkowski functionals for a toroidal[10] pore through an 8.8×8.8 nm^2 layer of 4.0 nm thickness as a function of the pore radius,[11] obtained using a grid size of 0.2 nm and by expanding the coordinates to a radius of 0.1 nm. In addition, the analytical values for the volume V, surface area A, and integrated mean curvature H are plotted:[12]

$$V = V_{\text{slab}} - V_{\text{cyl}} + \pi^2 d^2(d+r) - \frac{4}{3}\pi d^3, \tag{1}$$

$$A = 2(A_{\text{rec}} - A_{\text{circ}}) + 2\pi^2 d(d+r), \tag{2}$$

$$H = \pi^2(d+r) - 4\pi d. \tag{3}$$

In these expressions, d is half the thickness of the slab, r is the radius of the pore at its smallest extension, A_{rec} is the area of the bottom or top of the unporated slab, A_{circ} is the area of the circle with radius $d+r$, V_{slab} is the volume of the unporated slab, and V_{cyl} is the volume of the cylinder with a height of $2d$ and a radius of $d+r$.

---

[10] In other words, a pore in which the curvature of the surface varies smoothly in a manner identical to the "inner" part of a torus.
[11] Defined as the radius of the actual opening.
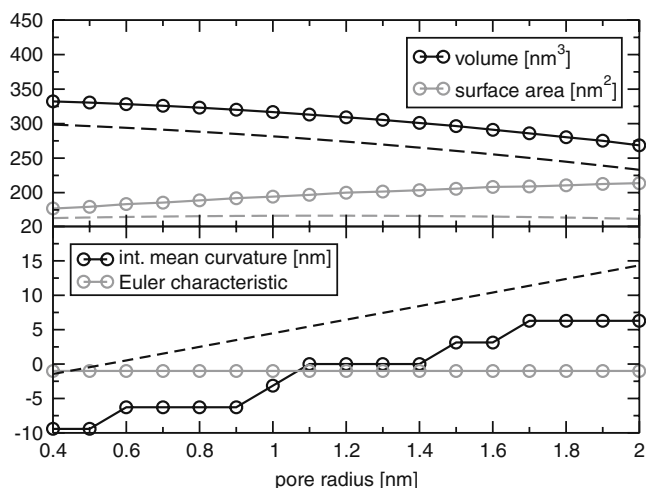[12] Note that periodic boundary conditions apply.

Fig. 7 Morphological image analysis of model systems of ideal toroidal pores of different radii (*lines between measured points* serve to guide the eye). Calculated values corresponding to the underlying geometry are plotted (*dashed lines*) for comparison

As before, the Euler characteristic is obtained with the exact value of −1, and the volume of the image is higher but proportional to that of the original. The surface area is overestimated to a larger extent, again showing how the area of a curved surface is increased by the rasterization of the image. The fact that the surface area of the image is actually found to increase over the whole range of radii, in contrast to the values calculated for the original, reflects the increasing percentage of the total surface that is curved for larger pore radii. This causes the slight decrease in the surface area in the original geometry to be overshadowed by the overestimation of areas of curved surfaces in the image.

The integrated mean curvature shows the same general trend for both image and original, but the amount of negative curvature is higher in the image for the measured range of radii. This causes small pores to display negative values for radii of up to 1 nm, while the actual crossover point for the original geometry occurs at approximately 0.5 nm. In addition, it becomes apparent that the values obtained by morphological image analysis are discrete and not continuous,[13] causing small changes in curvature in the original geometry to go unnoticed in the image.

The local curvatures were calculated using four orientations for each axis and by averaging over neighboring voxels up to a distance of five grid cells. Looking at the mapping onto the image shown in Fig. 8, the dominance of negative mean curvature for pores of small radii found in the global values is also visible. The mean curvature is accurately found to be minimal in the midsections of the

pores, reflecting the fact that the highest negative principal curvature is located in that region, and maximal close to the rim, reflecting the fact that the lowest negative principal curvature occurs in that region,[14] and is in fact accurately found to be approximately zero in the midsection of the pore of radius 2.0 nm (for this radius and a layer thickness of 4.0 nm, the two principal curvatures cancel in this region). In addition, it becomes more positive overall for higher pore radii, in accordance with the lower negative principal curvature. The Gaussian curvature is also found to be accurately mapped, with the maximum (negative) curvature found in the midsection, and the curvature gradually decreasing to zero the closer one gets to the rim for the two bigger pores. The minimum Gaussian curvature in the midsection is only not detected for the smallest radius, due to the pore size being close to the limit of the resolution used. In principle, this problem could be avoided by using a higher resolution.

It is worth mentioning that the negative spaces of the images of the ideal toroidal pores are images of a stalk. The corresponding stalks will therefore have identical Gaussian curvature and surface area to the pores, but the sign of the mean curvature will be inverted. For the global values, it can therefore be deduced that stalks are accurately characterized as having negative mean curvature if one considers that stalks have a certain minimum radius given by the lipid tail length (approximately 2.0 nm for a typical lipid tail of 16–18 carbon atoms).

Applications

Next we tested our method with trajectories and snapshots taken from actual simulations of lipids. For these, it proved advantageous to define the positive phase as only the atoms or beads corresponding to the lipid tails. This allowed details like pores to be amplified and stalks to be distinguished from configurations in which two membranes are close but there is no contact between the hydrophobic cores.

Spontaneous aggregation The first application is the determination of the phase adopted by a mixture of DOPE and water, starting from random coordinates, in a spontaneous aggregation approach [15]. The Minkowski functionals obtained using a grid size of 0.5 nm and by expanding the coordinates to a radius of 0.4 nm are shown in Fig. 9. Looking at the Euler characteristic, the most significant morphological indicator, one can see that the system quickly adopts a metastable phase, where it remains

---

[13] The distance between the discrete levels depends on the grid resolution.

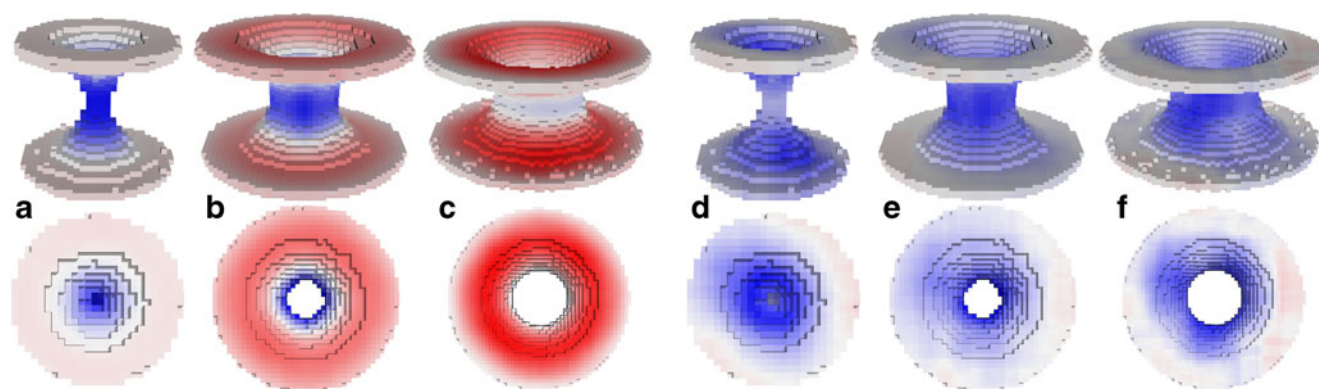[14] The second, positive, principal curvature is constant across the whole pore.

Fig. 8a–f Mapping the local values of mean (**a**, **b**, **c**) and Gaussian (**d**, **e**, **f**) curvature onto the constructed images of model systems of ideal toroidal pores. Shown are the results for pores with radii of

0.4 nm (**a**, **d**), 1.2 nm (**b**, **e**) and 2 nm (**c**, **f**). Positive curvature is depicted in *red*, zero curvature in *white*, and negative curvature in *blue*

for just over 0.5 µs, before it adopts its final configuration, in which it remains for the rest of the simulation. The metastable state has a negative Euler characteristic of −2, which, when compared to a lamellar state, suggests the presence of pores or stalks. Taking into account the amount of negative mean curvature, the most likely state is a stalk phase, since pores display a higher mean curvature. The stable phase adopted for the rest of the simulation has a Euler characteristic of 0, suggesting a lamellar or inverted hexagonal phase. The fact that the integrated mean curvature remains negative rules out the lamellar phase, leaving just the inverted hexagonal phase. Visual inspection confirms these findings.

Porated membrane As a second application, we looked at a closing pore in a DPPC membrane. The Minkowski functionals shown in Fig. 10 were obtained using a grid size of 0.5 nm and by expanding the coordinates to a radius of 0.4 nm. The closure of the pore can be detected and is clearly

reflected in the Euler characteristic, the integrated mean curvature as well as the surface area, with the observed changes agreeing with our earlier measurements for the model pores (see above). The presence of noise, especially in the integrated mean curvature and the surface area, stems from natural fluctuations in the coordinates as well as translational movements of the system and fluctuations in the box size, all of which cause changes in the image. However, averaging over time reduces the noise significantly, as demonstrated for the integrated mean curvature in Fig. 10.

In addition, we chose this application to demonstrate the effects of the chosen grid resolution and the radius used for the coordinate expansion on the results obtained. Table 2 shows the average values of the Minkowski functionals for the open pore (i.e., the first 30 nm of the trajectory analyzed in Fig. 10) using a higher resolution or a smaller expansion radius. The deviations of the Euler characteristic from the correct value of 1 indicate the presence of noise in the analysis. Considering the nature of the changed parameters, a
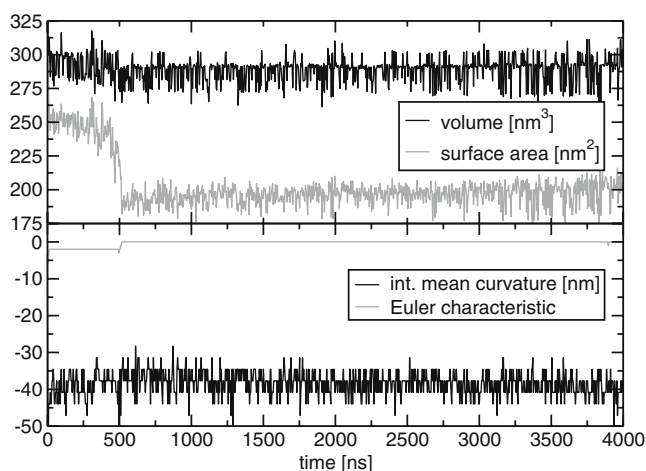

Fig. 9 Morphological image analysis of the spontaneous aggregation of a random mixture of lipids and water into an inverted hexagonal phase
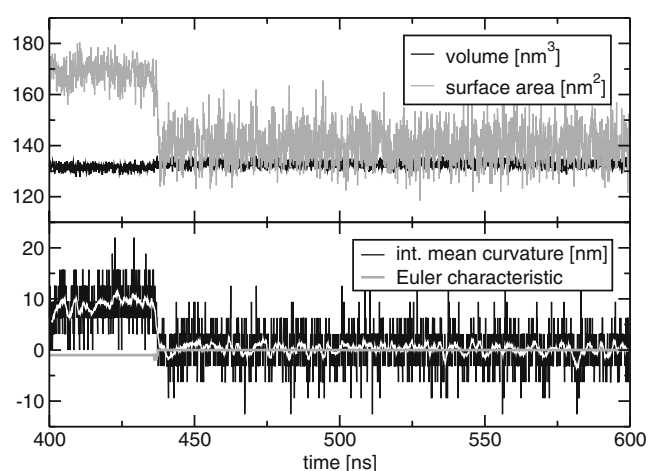

Fig. 10 Morphological image analysis of the trajectory of a closing pore. In addition to the properties indicated by the legend, the running average of the integrated mean curvature is plotted in *white*

Table 2 Average values of the volume V, the surface area A, the integrated mean curvature H, and the Euler characteristic χ extracted from the simulation of a porated membrane in relation to the resolution (identified by the edge length d of the grid) and the radius of the spherical cloud r_S used for the expansion of the spheres. Note that, in order to ensure that no effects are masked, cluster filtering was not applied to the images

r_S (nm)	d (nm)	V (nm^3)	A (nm^2)	H (nm)	χ
0.4	0.5	131.47(6)	168.8(2)	8.8(2)	−1.000(0)
0.4	0.4	117.4(1)	169.7(2)	11.4(2)	−0.89(2)
0.4	0.3	100.76(3)	247.2(3)	−205(1)	71.4(4)
0.3	0.5	126.00(6)	169.7(2)	11.3(2)	−0.997(3)
0.2	0.4	120.25(6)	170.1(3)	12.2(2)	−0.71(3)

reduced expansion radius or a higher grid resolution can both potentially lead to empty grid cells in regions occupied by the molecular aggregate. As a result, additional tunnels (which would lower the value by one in that particular frame of the trajectory) as well as cavities (which would raise the Euler characteristic in that respective frame by one) are both possible. However, the fact that the Euler characteristic increases indicates that cavities are the dominant artifact. The observed decrease in volume and the increase in surface area both corroborate this interpretation. The changes in the integrated mean curvature are less clear (except for the drastic change to negative values at a grid resolution of 3 nm), but, as has been pointed out in "Methods," the mean curvature obtained depends on the system's propensity to produce bumps and dents in the image, which in turn depends on the parameters used, making a comparison of the curvatures obtained with different parameters difficult.

The mapping of the local curvatures obtained for a snapshot of the open pore is depicted in Fig. 11. Data were used from eight grid orientations for each axis at a resolution of 0.3 nm, with the coordinates expanded to 0.3 nm and averaging performed over neighboring voxels up to a distance of three grid cells. The mean and Gaussian curvatures associated with the pore are accurately mapped as being positive and negative, respectively. While the distributions are less homogeneous than those for the ideal model systems, this is not an artifact of the method; it indicates the accurate detection of features that are present at the analyzed coordinates. Regions with groups of atoms protruding from the mean are correctly displayed as having high mean and Gaussian curvatures,[15] whereas regions with saddle-splayed surfaces are shown to have low mean and high (negative) Gaussian curvatures. The general trend seen with the model systems for the location of the highest local

---

[15] In fact, the Gaussian curvature in these regions should be positive, but this is likely masked because the surrounding negative Gaussian curvature is carried into the relatively small area of positive curvature by the averaging procedure.
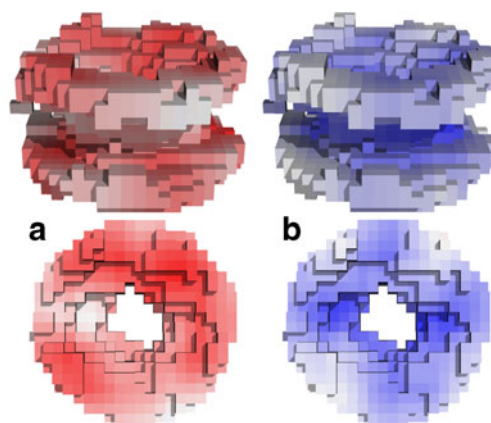

Fig. 11a–b Visualization of the local distributions of mean (**a**) and Gaussian (**b**) curvature for a snapshot of a pore in a DPPC membrane. Positive curvature is depicted in *red*, zero curvature in *white* and negative curvature in *blue*

mean curvature to be close to the rim of the pore and the highest Gaussian curvature to occur in the midsection is also preserved in the simulated pores, but is slightly modified by the superposition of effects due to deviations from the ideal toroidal shape.

For comparison, a snapshot of the underlying structure is shown in Fig. 12.

Stalk formation Figure 13 shows the Minkowski functionals obtained for the formation of stalks between two DOPE bilayers at low hydration and high temperature. Data were obtained using a grid resolution of 0.2 nm and expanding the coordinates to a radius of 0.34 nm. Starting
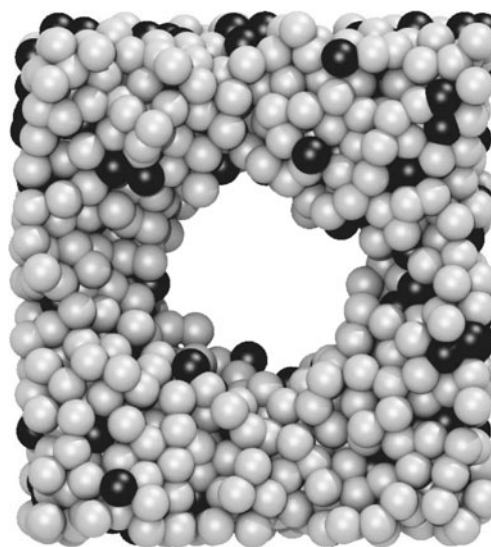

Fig. 12 Snapshot taken from the trajectory of a closing pore in a DPPC membrane. For clarity, only the beads representing the lipid carbon tails (the terminal beads are shown in *black*, the remaining beads in *gray*), which are the beads used to define the positive space in our analysis, are shown
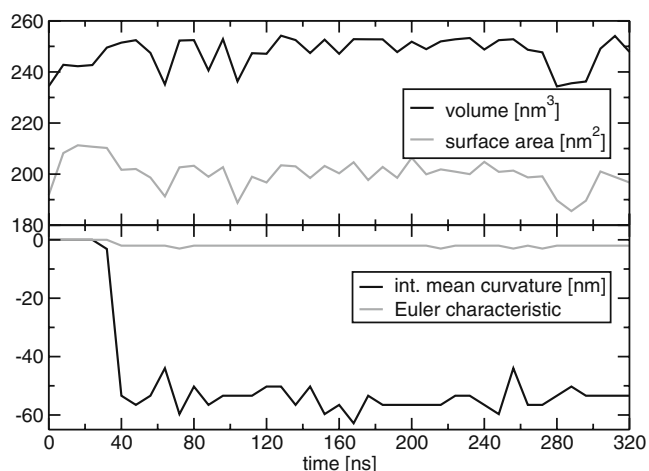
Fig. 13 Morphological image analysis for the simulation of stalk formation starting from two isolated bilayers
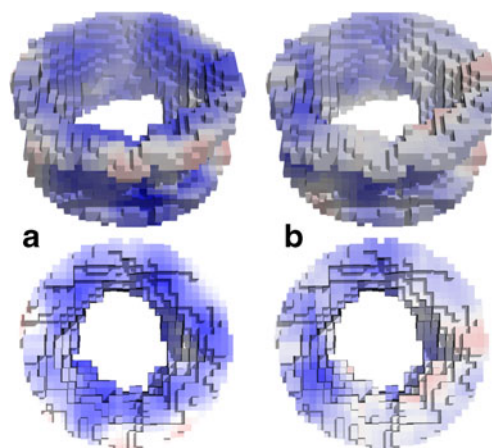

Fig. 14a–b Visualization of the local distributions of mean (**a**) and Gaussian (**b**) curvature for a snapshot of a stalk between two DOPE membranes. Positive curvature is depicted in *red*, zero curvature in *white*, and negative curvature in *blue*

from a lamellar configuration, the formation of the stalks is reflected in a change in the Euler characteristic from 0 for the two bilayers to −2, indicating the simultaneous formation of two stalks. At the same time, the integrated mean curvature drops from 0 to negative values, also indicating the formation of stalks.

Shown in Fig. 14 is the mapping of the local curvatures of an isolated stalk using data from four rotations of the grid around each axis at a resolution of 0.17 nm while expanding the coordinates to a radius of 0.34 nm and averaging over neighboring voxels up to a distance of three grid cells. As for the simulated pore, the detected curvature is not homogeneous, but the general trend in the relative distribution of curvature between foot (the counterpart to the rim of a pore) and midsection is also preserved here: a higher (negative) mean curvature is observed close to the foot of the stalk, and a higher (negative) Gaussian curvature close to the midsection. While some of the observed inhomogeneities, especially in regions of relatively low curvature, are likely caused by artificial noise introduced by insufficient averaging, most of the detected curvature can again be attributed to actual morphological properties present at the analyzed coordinates.

A snapshot of the formed stalk is shown in Fig. 15.

Computational costs

To give a rough indication of the time required to perform the analyses presented here, we will state the CPU times[16] needed for some of the performed calculations. This is not intended to be an extensive analysis of scaling and computational efficiency, but rather to act as an aid for readers interested in using the method. So far, the program

has not been optimized for computational efficiency; however, the time required to perform the presented morphological analyses is still on the order of only a few minutes.

To compute the global morphology for the trajectory of the closing pore shown in Fig. 10, a total CPU time of 120 s was needed for all 2,500 frames, corresponding to approximately 50 ms per frame. If the grid resolution is reduced by a factor of two, the CPU time required for the whole trajectory drops to 104 s.

To compute the local curvatures, the time required to analyze the largest of the model pores presented in Fig. 8 was 187s, corresponding to approximately 7 s per orientation
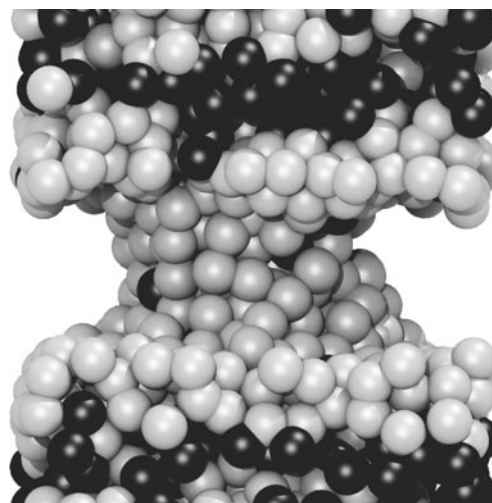

Fig. 15 Snapshot of a stalk formed between two apposed DOPE membranes. For clarity, only the beads representing the lipid carbon tails (the terminal beads are shown in *black*, the remaining beads in *gray*), which are the beads used to define the positive space in our analysis, are shown

---

[16] For an Intel Core 2 Duo 6700 2.66 GHz CPU.

used in the averaging. As before, reducing the grid resolution increases the performance, with a reduction in resolution of a factor of two decreasing the time required to 132 s. Note that the analysis of the local curvature for a snapshot of the pore taken from the simulation referred to in the previous paragraph is performed much more rapidly, due to the lower coordinate density and system size. The results shown in Fig. 11 required a CPU time of 57 s, corresponding to 167 ms per orientation used for the averaging.

Conclusions

Our tool uses the technique of morphological image analysis to analyze sets of coordinates that describe aggregates of soft matter. The implementation is intended for the analysis of mixtures of lipids and water obtained from molecular dynamics simulations, but is also in principle applicable to all kinds of coordinates that describe binary mixtures, and is independent of the model and the method used to generate the coordinates.

It has been demonstrated to be helpful for a range of morphological tasks, including phase detection and the monitoring of dynamic processes like stalk formation and pore closure. While the global values obtained for the Minkowski functionals are subject to the limitations inherent in the technique of morphological image analysis (i.e., overestimation of the surface area and the integrated mean curvature), the most significant value—the Euler characteristic—is obtained correctly and virtually free of noise. In addition, the systematic nature of the error in the total mean curvature means that it can be used in comparative analyses and to more generally detect the presence or absence of mean curvature, as well as to distinguish systems with negative from those with positive values, enabling a broad spectrum of applications. Only in complex systems that contain both stalk-like and pore-like structures simultaneously can one encounter problems with interpretation. In such ambiguous cases, analyzing the number of separate clusters for both components of the binary mixture—which can also be done in our implementation—can help.

In addition, by mapping the local values of mean and Gaussian curvature onto an image representing the system, morphological features that are overlooked during naive analysis by eye can be visualized. While the tool still exhibits some difficulties in avoiding artificial inhomogeneities during the detection of local curvature in larger areas with low homogeneous curvature (due to the rasterization of the underlying image), the general trend towards a specific curvature in a given area is preserved, and areas that show an actual prevalence of either positive or negative curvature in comparison with their neighborhoods are accurately visualized.

So far, our implementation only analyzes coordinate sets as they are generated by the simulation, and does not include an option for time averaging over multiple frames of a trajectory. If an analysis of average structures is desired, it is therefore necessary to create an averaged coordinate set before the analysis is performed.

We hope that our program, which is compatible with the popular Gromacs package, will allow a wide range of users to benefit from using morphological image analysis in their research.

References

1. Marrink SJ, de Vries AH, Tieleman DP (2009) Biochim Biophys Acta Biomembr 1788(1):149
2. Michielsen K, De Raedt H (2001) Phys Rep 347(6):461
3. Hadwiger H (1957) Vorlesungen über Inhalt, Oberflache und Isoperimetrie. Springer, Berlin
4. Loison C, Mareschal M, Schmid F (2004) J Chem Phys 121 (4):1890
5. Marrink SJ, Tieleman DP (2002) Biophys J 83(5):2386
6. Sevink GJA, Zvelindovsky AV (2005) Macromol 38(17):7502
7. Edelsbrunner H, Mücke E (1994) Assoc Comput Mach Trans Graph 13(1):43
8. Zomorodian A, Carlsson G (2005) Discrete Comput Geom 33 (2):249
9. Kasson PM, Zomorodian A, Park S, Singhal N, Guibas LJ, Pande VS (2007) Bioinformatics 23(14):1753
10. van der Spoel D, Lindahl E, Hess B, Groenhof G, Mark AE, Berendsen HJC (2005) J Comput Chem 26(16):1701
11. Hyde ST, Barnes IS, Ninham BW (1990) Langmuir 6(6):1055
12. DeLano WL (2002) The PyMOL molecular graphics system. DeLano Scientific, San Carlos
13. Marrink SJ, Risselada HJ, Yefimov S, Tieleman DP, de Vries AH (2007) J Phys Chem B 111(27):7812
14. Berendsen HJC, Postma JPM, DiNola A, Haak JR (1984) J Chem Phys 81:3684
15. Marrink SJ, Lindahl E, Edholm O, Mark AE (2001) J Am Chem Soc 123(35):8638
16. Leontiadou H, Mark AE, Marrink SJ (2004) Biophys J 86(4): 2156
17. Marrink SJ, Mark AE (2004) Biophys J 87(6):3894