

Supplemental material for

“Pattern-wavelength coarsening from topological dynamics in silicon nanofoams”

M. Castro, R. Cuerno, M. M. García-Hernández, and L. Vázquez

1 Algorithmic extraction of topological information

As mentioned in the main text (MT), the main difference between our approach and previous ones is that we compute directly the area of every *cell*, rather than estimating it using e.g. a Voronoi tessellation.

This involves several steps that allow to replace the original AFM image by an adjacency matrix among cells (M_{ij}). The latter moreover provides the relevant topological information on the cell arrangement that is further analyzed in the MT. In what follows we illustrate such a processing protocol. The same procedure has been applied both to the experimental and to the numerical images obtained from simulations of Eq. (6) in the MT. Overall, taking into account the fact that we need to study the evolution of the system in time, this has required processing of a large number of images (up to 1000 in the case of the simulations, taking into account that averages over initial conditions are required at each fixed time).

1.1 Image processing

The extraction of the information from an AFM image or from a simulation morphology involves its transformation and further recombination with some derivative images. The goal is to determine clearly the boundary of each cell, in order to further study its geometrical (area) and topological properties within the arrangement. The transformations we employ are standard in the context of so-called morphological image analysis [1], and have been implemented through straightforward use of built-in commands of the Matlab image processing toolbox. The resulting figure is a *posterized* version of the original one and is amenable to quantitative analysis of the type described in the MT.

The process we have followed involves the following steps, exemplified in Fig. 1:

1. Top-hat and bottom-hat transforms: These extract the areas in the image which have highest or lowest brightness, respectively.
2. Re-combination of the original image and the two images obtained after Step 1: By further adding/subtracting the top-hat and bottom-hat transforms, respectively, to/from the original image, boundaries among cells are emphasized, as better appreciated after the next step.
3. Complementation of the image in Step 2: As the inner region of the cells does not contain relevant information in this context, the *negative* image of that after Step 2 is more convenient to process. Notice the enhancement of inter-cell boundaries already achieved after Step 3, as compared with the initial image.
4. Detection of intensity valleys: This step allows, precisely, to remove the deepest regions within each cell. These *valleys* will be removed to *flatten* the region within each cell in the next Step.

5. Watershed segmentation: By defining a threshold *height*, we can filter the image obtained by subtracting those after Steps 3 and 4. The idea is that, after the present processing Step, only the boundaries are *bright enough*, so that all the information below that threshold has been removed. The resulting image is a collection of regions with value 0 (below the threshold) and boundaries with value 1.
6. Cell color and number assignment: Using a standard *color filling* algorithm, we can assign a color to each cell. Then, through the color map, a different number is assigned to each individual cell.

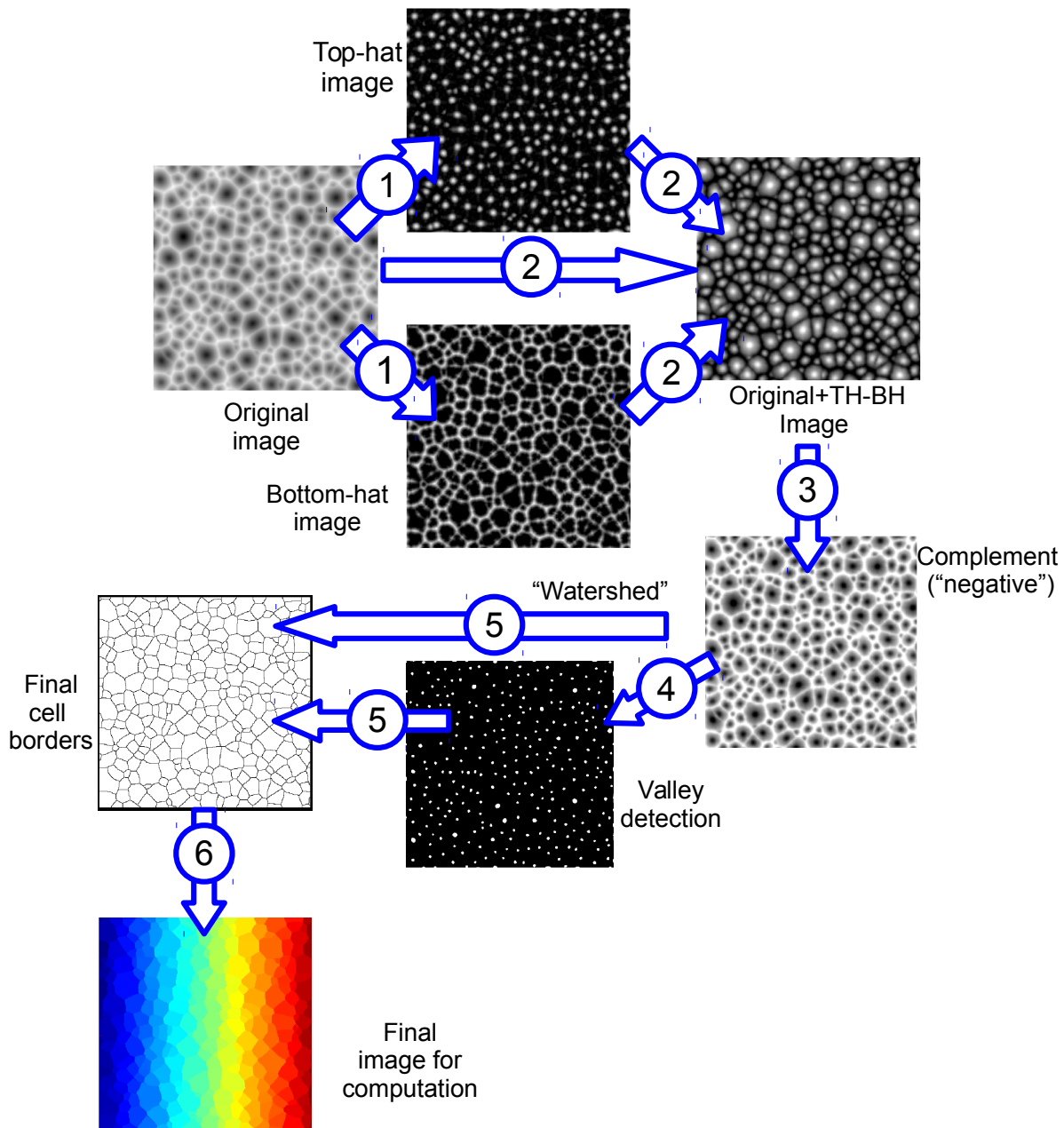


Figure 1: Schematic summary of the image processing steps performed to extract the topological properties from the cellular patterns. Although many grains seem to have the same color, it is due to a lack of resolution not to an overlapping in the grain numbering.

The cell boundaries can be explicitly extracted to illustrate the fidelity of the algorithm with

respect to the original sample (Fig. 2).

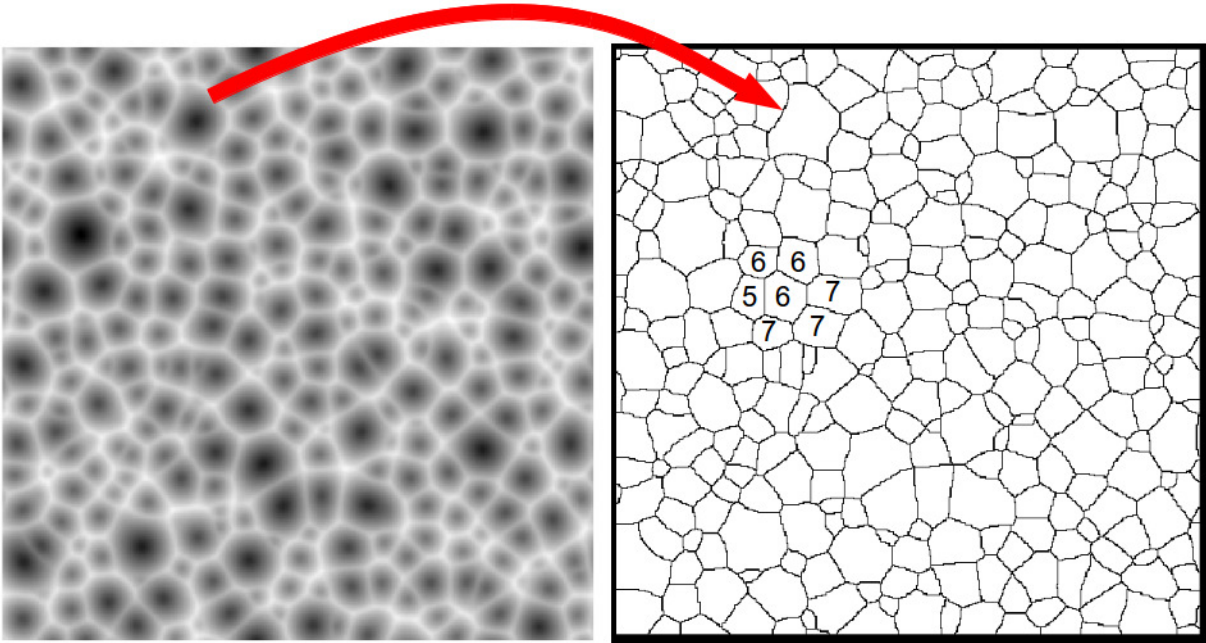


Figure 2: Left: Same original image as in Fig. 1. Image size is $50 \times 50 \mu\text{m}^2$. Right: Cell boundaries obtained following the steps summarized in Fig. 1. The right arrow shows how a cell in the original image corresponds to a region in the processed one. Numbers of neighbors, as computed by our algorithm, are shown for a few sample cells.

1.2 Adjacency matrix and information on the cellular pattern

At the last step in Sec. 1.1, a number is assigned to each *color* (representing each a specific cell in the pattern). After removal of regions that impinge the boundaries of the image, we find an adjacency matrix, $M \in \mathbb{R}^{N \times N}$, where N is the total number of remaining cells.

The process of assigning an adjacency matrix to the final image after Step 6 in the processing protocol described above is straightforward: we run two loops over coordinates x and y through the image, where $x, y = 1, \dots, \text{Pixel} \#$, where the lateral number of pixels is typically $\text{Pixel} \# = 512$. The simulations are scaled so that $\text{Pixel} \#$ is 512 as well. At every pixel position (x, y) (that corresponds, say, to the specific cell with color number i), we determine the color of the 4 neighboring pixels ($x \pm 1$ and $y \pm 1$). If the color of a nearest neighbor is different from the present one and corresponds to color $j \neq i$, we set $M_{ij} = M_{ji} = 1$. A typical form of the matrix looks like

$$M = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 1 & 0 & 0 & \dots \\ 0 & 0 & 1 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 1 & \dots \\ 0 & 0 & 0 & 0 & 1 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

As stated, $M_{ij} = 1$ means that cell i is adjacent to cell j , with $i, j = 1, \dots, N$, thus M is symmetric. The x value of diagonal elements remains arbitrary.

Additional observables, like e.g. the number of sides for each grain, can be obtained from M , in this case by summing all non-diagonal elements in the corresponding row of the adjacency

matrix. Furthermore, also from the *colored* image¹ in Fig. 1, we can count the number of pixels assigned to each cell. By rescaling that number to the whole physical area of the experiment, we can convert from “*number of pixels*” to μm^2 .

This automatic procedure allows to process large amounts of information, in particular from the numerical simulations of the dynamics described by the cCH equation of the main text. Hence, we can follow the evolution of observables in time, as shown in the figures of the main text (mean cell area, entropy, number of neighbors, etc.). As an example, in Fig. 3 we show the histogram of areas corresponding to Fig. 2.

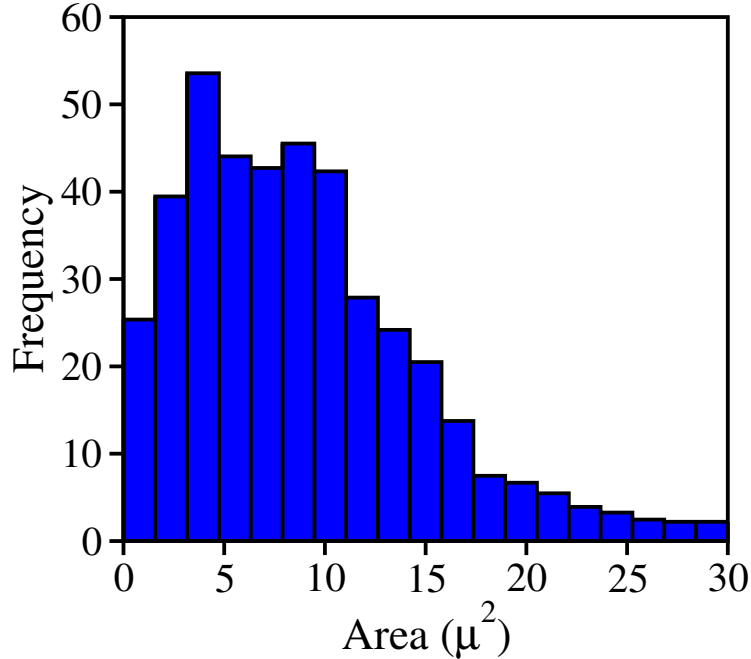


Figure 3: Histogram of physical areas obtained from Fig. 2.

2 Statistical significance of the fits

In the main text, we show the agreement between our data and several topological laws, such as those of Aboav, Lewis, or Lemaitre. To assess the goodness of the linear fits provided (in particular in the case of Figs. 2a,b of the main text), we have evaluated the correlation coefficient and the so-called reduced- χ^2 coefficient [2], defined as

$$\text{reduced-}\chi^2 = \frac{1}{\nu - m} \sum_{i=1}^{\nu} \left(\frac{y_i - \bar{y}_i}{\sigma_i} \right)^2,$$

where y_i is the experimentally observed value, with error bar σ_i , and \bar{y}_i is the fitted one. The parameter ν stands for the number of measured values from the experiment and m for the number of fitting parameters ($m = 2$ in the case of a linear regression). The χ^2 test provides an objective quantification of the goodness of the fit. In particular, values of the reduced- χ^2 between 0.1 and 10 are considered indicative of meaningful fits.

Acknowledgments

¹For computational performance, this is done within the same loops employed in the creation of the adjacency matrix.

We thank Raúl Gago for discussions. This work has been partially supported by MICINN and by MEC (Spain), through Grants Nos. FIS2009-12964-C05-01, FIS2009-12964-C05-03, FIS2009-12964-C05-04, FIS2012-38866-C05-01, and FIS2012-38866-C05-01-05.

References

- [1] P. Soille, *Morphological image analysis: principles and applications* (Springer-Verlag, New York, 2003).
- [2] P. R. Bevington and D. K. Robinson, *Data reduction and error analysis for the physical sciences (Vol. 2)* (McGraw-Hill, New York, 1969).