



Segmentation of scanning tunneling microscopy images using variational methods and empirical wavelets

Kevin Bui¹ · Jacob Fauman² · David Kes³ · Leticia Torres Mandiola⁴ · Adina Ciomaga⁵ · Ricardo Salazar⁶ · Andrea L. Bertozzi⁶ · Jérôme Gilles⁷ · Dominic P. Goronzy^{8,9} · Andrew I. Guttentag^{8,9} · Paul S. Weiss^{8,9,10}

Received: 7 June 2018 / Accepted: 22 April 2019
© Springer-Verlag London Ltd., part of Springer Nature 2019

Abstract

In the fields of nanoscience and nanotechnology, it is important to be able to functionalize surfaces chemically for a wide variety of applications. Scanning tunneling microscopes (STMs) are important instruments in this area used to measure the surface structure and chemistry with better than molecular resolution. Self-assembly is frequently used to create monolayers that redefine the surface chemistry in just a single-molecule-thick layer (Love et al. in *Chem Rev* 105(4):1103–1170, 2005; Nuzzo and Allara in *J Am Chem Soc* 105(13):4481–4483, 1983; Smith et al. in *Prog Surf Sci* 75(1):1–68, 2004). Indeed, STM images reveal rich information about the structure of self-assembled monolayers since they convey chemical and physical properties of the studied material. In order to assist in and to enhance the analysis of STM and other images (Thomas et al. in *ACS Nano* 10(5):5446–5451, 2016; Thomas et al. in *ACS Nano* 9(5):4734–4742, 2015), we propose and demonstrate an image processing framework that produces two image segmentations: One is based on intensities (apparent heights in STM images) and the other is based on textural patterns. The proposed framework begins with a cartoon + texture decomposition, which separates an image into its cartoon and texture components. Afterward, the cartoon image is segmented by a modified multiphase version of the local Chan–Vese model (Wang et al. in *Pattern Recognit* 43(3):603–618, 2010), while the texture image is segmented by a combination of 2D empirical wavelet transform and a clustering algorithm. Overall, our proposed framework contains several new features, specifically in presenting a new application of cartoon + texture decomposition and of the empirical wavelet transforms and in developing a specialized framework to segment STM images and other data. To demonstrate the potential of our approach, we apply it to raw STM images of various monolayers and present their corresponding segmentation results.

Keywords Scanning tunneling microscopy · Segmentation · Chan–Vese · Empirical wavelets · Textures

1 Introduction

Self-assembled monolayers (SAMs) have been extensively studied and applied in nanoscience, nanotechnology, and beyond [34, 43, 56, 59]. These SAMs are formed by molecules that have a head group (e.g., sulfur, selenium, carboxylate, phosphonate) that is chemically bound to a substrate (e.g., gold, silver, copper, platinum, germanium), and often form two-dimensional crystalline lattices (see Fig. 1). The strong affinity between the head group and the substrate and the intermolecular interactions between the molecular backbones and tail groups lead to self-assembly of the

monolayers via exposure of the substrate to the molecules in solution, in vapor, or by contact with a supporting structure such as a polymer stamp. The organization of the monolayer structure depends on the chemical properties and structures of the molecular monolayer components [17]. Controlling the design of a SAM requires properly tuning the chemical and physical properties of the assembled molecules [18]. As a result, control of basic parameters and external stimuli (e.g., deposition conditions, temperature, electrochemical potential, and illumination) on SAMs has been examined to target specific assemblies for nanotechnology research and applications [35]. One method of analyzing the chemical and physical properties of SAM is by examining molecular resolution images obtained by scanning tunneling microscopy.

In order to obtain a scanning tunneling microscopy (STM) image of a SAM, an atomically sharp conducting probe tip

✉ Kevin Bui
kbui1993@gmail.com

Extended author information available on the last page of the article

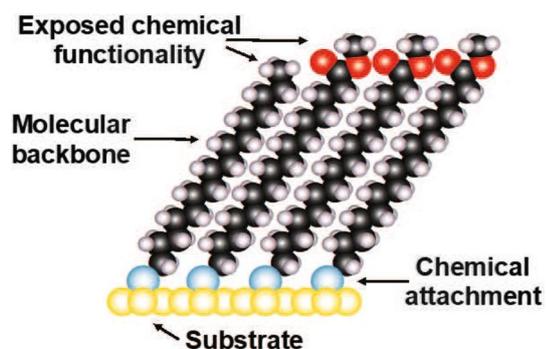


Fig. 1 In self-assembled monolayers, a single layer of molecules is chemically bound to a solid or liquid substrate. The wide range of substrates (e.g., metals, semiconductors, insulators, glasses, superconductors, nanoparticles) that can be used call for complementary chemistries of attachment of the molecular layers. The exposed functional group at the ends of the molecules typically dominates the interactions of the substrate with the surrounding chemical, physical, and biological environment

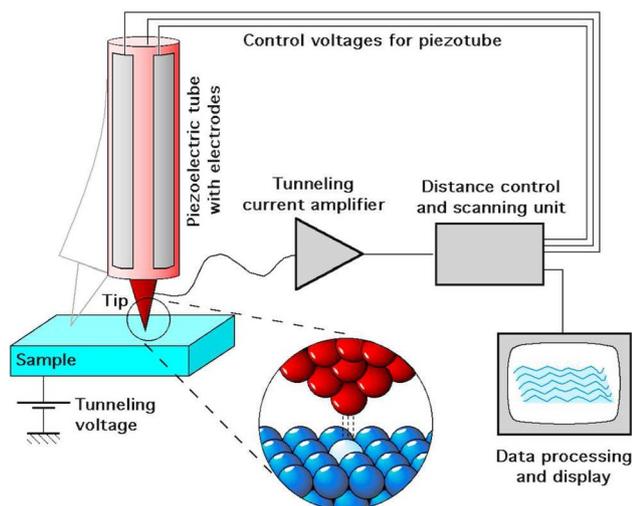


Fig. 2 To obtain scanning tunneling microscope images, constant current is held as the tip moves across the surface, experiencing voltage drops over bumps. Figure provided by Michael Schmid, TU Wien, at <https://commons.wikimedia.org/w/index.php?curid=180388>

is brought within one or two atomic diameters of the surface of the sample so that electrons can tunnel from the surface to the tip. A voltage bias is applied between the two, and the tip–sample separation is typically adjusted while scanning to maintain a constant tunneling current of electrons. Since the current is extremely sensitive to the tip–sample separation, better than atomic resolution is often obtained and apparent height differences across the surface are recorded, thereby acquiring nanoscale images with molecular features. The scanning procedure is shown in Fig. 2, and examples of STM images of various monolayers on Au{111} are shown in

Figs. 3, 4. These images show the varying textures and different apparent heights (displayed as intensities) as a result of the structure and chemical properties of the SAM. Partitioning the images according to the apparent heights and texture patterns would help facilitate the understanding and analyses of SAMs and other surfaces studied. We note that not only are the ordered regions important but also are the boundaries between them [56] since these domain boundaries determine access of other molecules to the substrate and can be used to isolate single molecules, or pairs, lines, or clusters of molecules [11, 17, 40].

Here, we propose a novel framework to analyze STM images that produces segmentation based on intensities and segmentation based on texture features. The proposed method consists of three main steps. The first step performs a cartoon + texture decomposition of the STM image. Its cartoon and texture components are then analyzed separately. The cartoon image is the component containing only the edges or boundaries of homogeneous regions in the images and it is devoid of any oscillatory patterns. On the other hand, the texture image consists of oscillatory patterns. We propose two parallel steps: The cartoon component is segmented using a variant of the Chan–Vese model [16] and the texture component is segmented by feeding some classifier with features based on empirical wavelets [30, 32].

First, we propose a novel multiphase version of the local Chan–Vese model [67] and develop an efficient algorithm based on the MBO scheme [46, 47] to solve it. The proposed model creates a more accurate segmentation result that is robust against intensity inhomogeneities compared to the original multiphase Chan–Vese model [66]. Secondly, we improve the 2D empirical wavelet transform, specifically the empirical curvelet transform, so that it provides a filter bank of empirical curvelets extracting meaningful textural information. From the filter bank outputs, we design a texture feature matrix to feed a clustering algorithm in order to identify regions of different textural patterns on the STM image.

The paper outline is as follows. In Sect. 2, we describe the nonlinear decomposition algorithm we use in our approach to decompose an STM image into its cartoon and textures components. In Sect. 3, we propose a modified version of the local multiphase model and determine its diffuse interface approximation in order to develop an algorithm based on the MBO scheme to solve it. In Sect. 4, we review the empirical wavelet transform (particularly the empirical curvelet transform) and propose modifications on how the curvelets' supports are detected. We then explain the design of the texture feature matrix based on the empirical curvelet coefficients. In Sect. 5, our proposed framework is applied to the images in Figs. 3, 4 in order to evaluate its efficacy in partitioning STM images by intensities and texture patterns. Finally, in Sect. 6, we summarize our results and discuss possible research directions to further improve the proposed framework.

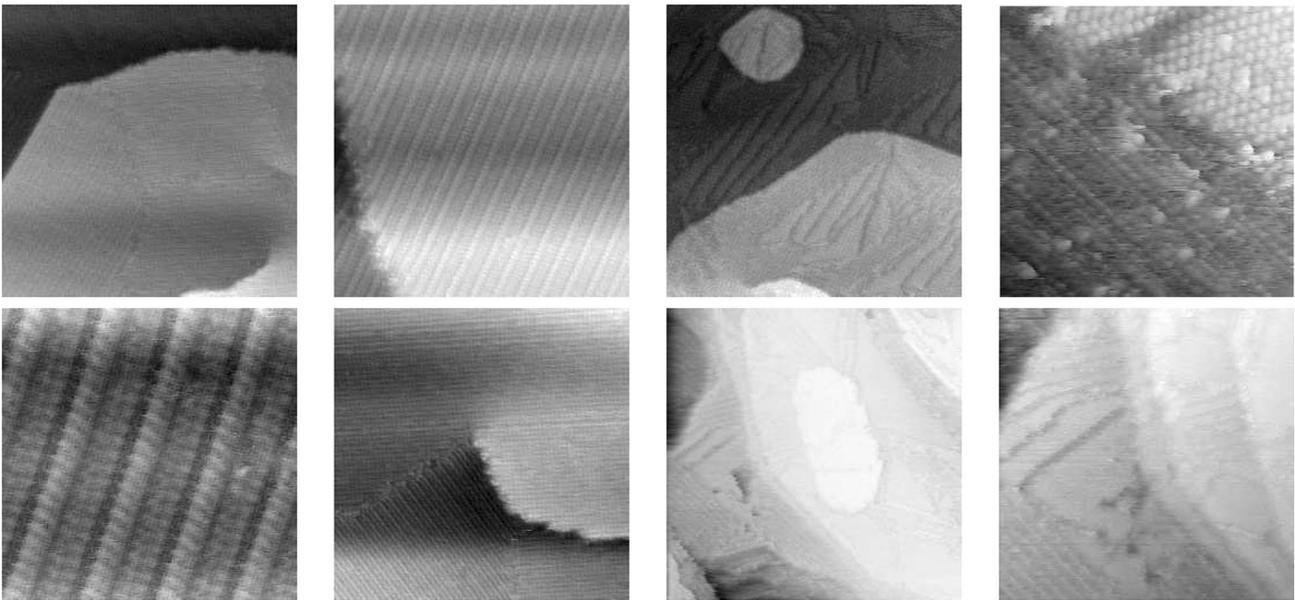


Fig. 3 Raw scanning tunneling microscope images of cyanide on Au{111}, reproduced from [35] with permissions. Images copyright American Chemical Society

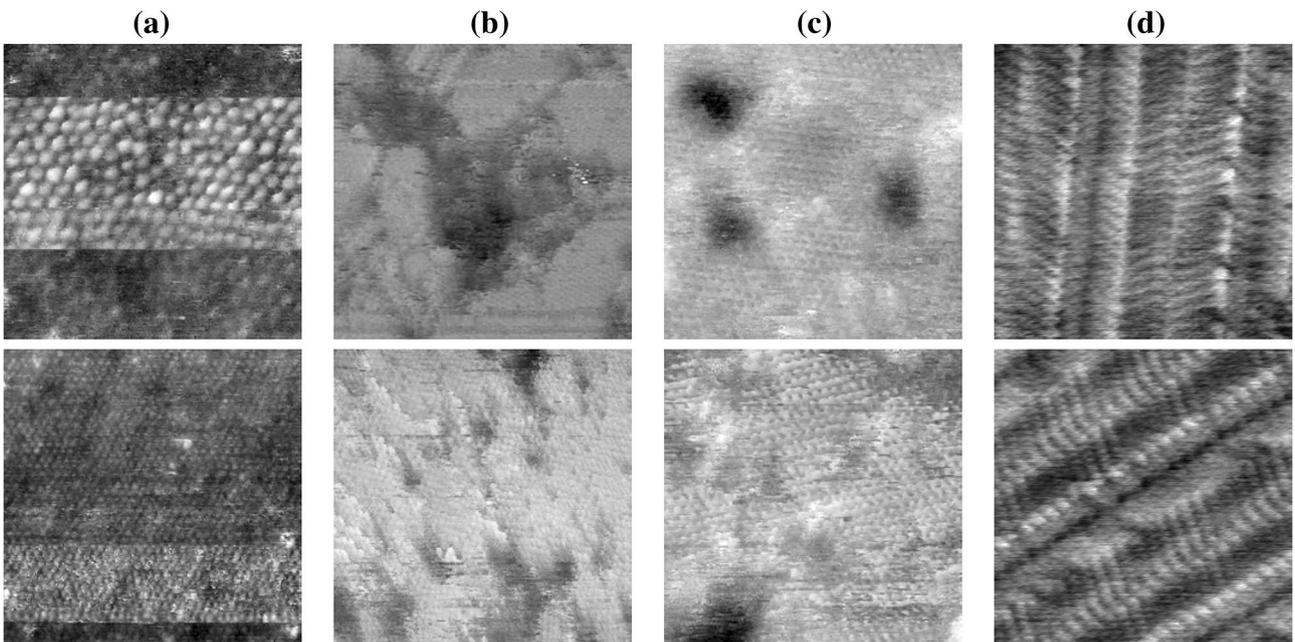


Fig. 4 Raw scanning tunneling microscope images of various monolayers. **a** 9,12-carboranedithiol on Au{111}/mica, see [62] for details; **(b)** holey graphene on Au{111}/mica, see [28] for details; **(c)** 3-mercaptop-*N*-nonylpropionamide (IATC9) on Au{111}, see [41,

61] for details; **(d)** peptides on highly oriented pyrolytic graphite, reproduced from [69] with permissions. Images copyright American Chemical Society

2 The cartoon + texture decomposition

As discussed in Introduction, the first step of our algorithm involves the decomposition of an image into its cartoon and textural components.

The cartoon + texture decomposition can be posed as an inverse problem and it consists of decomposing an image $f : \Omega \rightarrow \mathbb{R}$, where Ω is the image domain, into

$$f = u + v,$$

where u and v are the cartoon and the texture components, respectively. The cartoon image carries broad information about the image and is usually modeled by a function of bounded variation (piecewise smooth with possibly a discontinuity set). The texture image contains oscillatory information and is thus usually modeled by oscillating functions.

One of the earliest variational models that inspired cartoon + texture decomposition is the Rudin–Osher–Fatemi total variation minimization model [57]

$$\inf_{\substack{u \in BV(\Omega) \\ v \in L^2(\Omega)}} \left\{ \sigma \|u\|_{TV} + \|v\|_{L^2}^2, f = u + v \right\}, \tag{1}$$

where $\|u\|_{TV}$ is the total variation in u and σ is a tuning parameter that controls the regularization strength. The model was originally used for denoising purposes because of the functional spaces that u and v belong to. The function u belongs to the space of functions of bounded variations

$$BV(\Omega) = \left\{ u \in L^1(\Omega) : \int_{\Omega} |\nabla u| < \infty \right\},$$

which penalizes oscillations such as noise and textures but allows for piecewise smooth functions made of homogeneous regions with sharp boundaries. However, the decomposition is well posed only in a multiresolution setup since image features can be considered as texture in one scale and cartoon at a different scale. One of the most popular models in cartoon + texture decomposition is the $TV - L^1$ model proposed in [15], for which [14] provided a fast projection algorithm.

Although models, such as (1) and the above-mentioned, are able to perform cartoon + texture decomposition, Meyer argued that the texture image extracted by these models does not fully characterize the oscillatory patterns of the original image [48]. As a result, he proposed to replace the L^2 norm in (1) by weaker norms (associated with spaces of oscillatory distributions) in order to better capture the oscillatory patterns. In practice, some of these norms are difficult to compute. To remedy this drawback, Buades et al. developed a nonlinear version of the linearized Meyer’s model [8].

The linearized Meyer’s model is

$$\inf_{\substack{u \in H^1(\Omega) \\ v \in H^{-1}(\Omega)}} \left\{ \sigma^4 \|\nabla u\|_{L^2}^2 + \|v\|_{H^{-1}}^2, f = u + v \right\}, \tag{2}$$

where $H^1(\Omega) = \{u \in L^2(\Omega) : \nabla u \in L^2(\Omega)\}$ and $H^{-1}(\Omega)$ is the dual space of the homogeneous version of $H^1(\Omega)$. They are defined in the Fourier domain by

$$\begin{aligned} H^1(\Omega) \\ = \{u \in L^2(\Omega) : \int_{\Omega} [1 + (2\pi|\xi|)^2] |\mathcal{F}_2(u)(\xi)|^2 d\xi < \infty \} \end{aligned}$$

and

$$\begin{aligned} H^{-1}(\Omega) \\ = \{v \in L^2(\Omega) : \int_{\Omega} [1 + (2\pi|\xi|)^2]^{-1} |\mathcal{F}_2(v)(\xi)|^2 d\xi < \infty \}, \end{aligned}$$

where $\mathcal{F}_2(\cdot)$ is the 2D Fourier transform. Minimizing the quadratic functional (2) yields the solution

$$u = L_{\sigma} * f \text{ with } \mathcal{F}_2(L_{\sigma})(\xi) = \frac{1}{1 + (2\pi\sigma|\xi|)^4},$$

where the convolution kernel is given in the Fourier domain. Since $\mathcal{F}_2(L_{\sigma})(\xi)$ defines a low-pass filter, any frequency ξ that is significantly smaller than $\frac{1}{2\pi\sigma}$ is kept in u . Otherwise, it is kept in v . Thus, the cartoon + texture decomposition is performed by applying a low-pass/high-pass filter onto f ,

$$(u, v) = (L_{\sigma} * f, (Id - L_{\sigma}) * f).$$

The drawback of the low-pass filter is that sharp edges are altered, while they should be preserved as much as possible since they are necessary for cartoon segmentation.

The nonlinear version of (2) relies on a classifier which determines whether a pixel of the original image belongs to the cartoon or the texture component [8]. The idea consists of measuring the rate of change in the local total variation (LTV) between the original image and its low-pass filtered version, defined by

$$\lambda_{\sigma}(x) = \frac{LTV_{\sigma}(f)(x) - LTV_{\sigma}(L_{\sigma} * f)(x)}{LTV_{\sigma}(f)(x)}, \tag{3}$$

where

$$LTV_{\sigma}(f)(x) = L_{\sigma} * |\nabla f|(x). \tag{4}$$

If a neighborhood of a pixel x does not contain any textures, then f and $L_{\sigma} * f$ will be similar within the neighborhood, so $\lambda_{\sigma}(x)$ is close to 0. If some textures are present within the neighborhood, then the total variation in the neighborhood in the filtered image will be smaller and $\lambda_{\sigma}(x)$ is close to 1. Therefore, the cartoon component u is computed as the weighted average between the original image f and the filtered image $L_{\sigma} * f$ depending on λ_{σ} , i.e.,

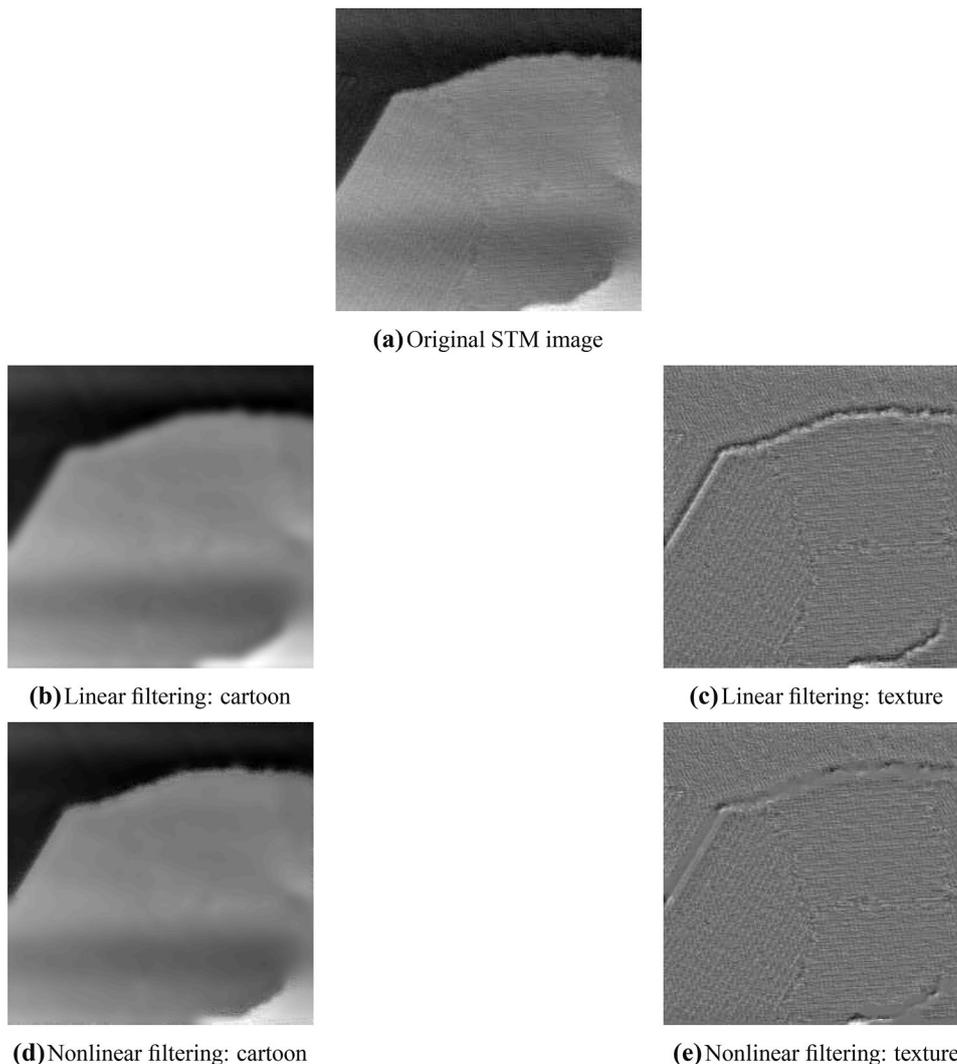
$$u(x) = w(\lambda_{\sigma}(x))(L_{\sigma} * f)(x) + (1 - w(\lambda_{\sigma}(x)))f(x),$$

with the soft threshold function $w : [0, 1] \rightarrow [0, 1]$ given by

$$w(x) = \begin{cases} 0 & \text{if } x < a_1 \\ (x - a_1)/(a_2 - a_1) & \text{if } a_1 \leq x \leq a_2, \\ 1 & \text{if } a_2 < x \end{cases}$$

where $0 < a_1 < a_2 < 1$. For our experiments in Sect. 5, as suggested in [8], we fixed $a_1 = 0.25$ and $a_2 = 0.50$. The texture component is easily obtained by computing the difference between the original image and its cartoon component.

Fig. 5 Results of linear and nonlinear cartoon texture decompositions with $\sigma = 3$, reproduced from [35] with permission. Image in (a) copyright American Chemical Society



The algorithm for the nonlinear version is summarized in Algorithm 1.

Algorithm 1: Cartoon+Texture Decomposition

Input: Original image f , parameter $\sigma > 0$

- 1: Compute the LTV reduction rate using (3) at each pixel.
- 2: Obtain the cartoon component:

$$u(x) \leftarrow w(\lambda_\sigma(x))(L_\sigma * f)(x) + (1 - w(\lambda_\sigma(x)))f(x)$$

- 3: Obtain the texture component:

$$v(x) \leftarrow f(x) - u(x)$$

Output: Cartoon and texture components u, v

In Fig. 5, we present results for both linear and nonlinear decompositions for one of the images in Fig. 3 to illustrate why choosing the nonlinear version is more interesting. Although both decompositions succeed in separating the image into its cartoon and texture components, as shown in

Fig 5b, d, the nonlinear decomposition gives sharper edges in the cartoon component than does the linear decomposition. Since in general these edges do not correspond to textures associated with molecular orientations but rather with topographic transitions of the molecular layers, the nonlinear version is therefore preferable.

3 Segmentation of the cartoon image

The cartoon component of an STM image provides information about the stratification of scanned molecules. The lighter the gray intensity is, the greater the apparent height. Here, the largest variations are due to monoatomic steps in the substrate; a single layer of molecules is present everywhere on the surface. Identifying different gray-level regions is therefore crucial for finding the topographic properties of the substrate and chemical layers.

We propose to use a level set approach based on the multiphase Chan–Vese (multiphase CV) model to perform the segmentation of the cartoon component. The Chan–Vese model is inspired by the pioneering work of Mumford and Shah [51], who suggested performing image segmentation by solving the minimization problem

$$\inf_{u, \Gamma} \left\{ \int_{\Omega \setminus \Gamma} |\nabla u|^2 + \lambda \int_{\Omega} (u - u_0)^2 + \mu \int_{\Gamma} ds \right\}, \tag{5}$$

where $u_0 : \Omega \rightarrow \mathbb{R}$ is the cartoon image to be segmented and μ and λ are weighing parameters. The function u is a piecewise smooth approximation of u_0 , which is allowed to have jumps across Γ —a closed subset of Ω given by a finite union of rectifiable curves.

Although several algorithms have been proposed to compute the solution [1], its computation [44, 65] is relatively complicated and numerically expensive. To overcome this drawback, simplifications of the energy functional (5) to piecewise constant functions have been proposed.

Based on the level set approach [54], the multiphase CV segmentation model [66] remains one of the most popular models. In this setup, the function u is allowed to have only four values c_1, c_2, c_3 , and c_4 , one for each of the four distinct regions. Furthermore, using two level set functions ϕ_1 and ϕ_2 , thus producing four phases $\{\phi_1 > 0, \phi_2 > 0\}$, $\{\phi_1 > 0, \phi_2 < 0\}$, $\{\phi_1 < 0, \phi_2 > 0\}$, and $\{\phi_1 < 0, \phi_2 < 0\}$, one can show it is sufficient to generate a partition of the domain Ω into regions of different intensities having triple junctions or T -junctions such that $\Gamma = \{\phi_1 = 0\} \cup \{\phi_2 = 0\}$ (see [66] for details).

Let $\mathbf{u} = (u_1, u_2)$ be a vector-valued function and $\mathbf{c} = (c_1, c_2, c_3, c_4)$ be a vector of constants. The fidelity term is defined as

$$\begin{aligned} \mathcal{E}_{fid}(\mathbf{c}, \mathbf{u}) &= \int_{\Omega} (c_1 - u_0)^2 u_1 u_2 \\ &+ \int_{\Omega} (c_2 - u_0)^2 u_1 (1 - u_2) \\ &+ \int_{\Omega} (c_3 - u_0)^2 (1 - u_1) u_2 \\ &+ \int_{\Omega} (c_4 - u_0)^2 (1 - u_1)(1 - u_2). \end{aligned} \tag{6}$$

The perimeter term is defined by

$$\mathcal{E}_{per}(\mathbf{u}) = \int_{\Omega} |\nabla u_1| + \int_{\Omega} |\nabla u_2|. \tag{7}$$

Using the standard Heaviside function H with the notations $\Phi = (\phi_1, \phi_2)$ and $H(\Phi) = (H(\phi_1), H(\phi_2))$, the four-phase

piecewise constant Mumford–Shah model can be written in terms of the level set functions ϕ_1 and ϕ_2 as

$$\mathcal{E}_{CV}(\mathbf{c}, H(\Phi)) = \lambda \mathcal{E}_{fid}(\mathbf{c}, H(\Phi)) + \mu \mathcal{E}_{per}(H(\Phi)). \tag{8}$$

3.1 Local multiphase Chan–Vese model

Model (8) is, however, not robust against illumination bias, such as shadows, or intensity inhomogeneities, such as weak edges, usually present in STM images. For example, if a region of an image is partially overlapped by a shadow, it might be segmented into two regions. Or, on the contrary, regions with inhomogeneous gray-level intensities may not be segmented at all, even though they might provide valuable information.

Wang et al. proposed a local term that can be added to (8) to counteract the lighting issues [67]. We define, as before, a local term

$$\begin{aligned} \mathcal{E}_{loc}(\mathbf{d}, \mathbf{u}) &= \int_{\Omega} (g_k * u_0 - u_0 - d_1)^2 u_1 u_2 \\ &+ \int_{\Omega} (g_k * u_0 - u_0 - d_2)^2 u_1 (1 - u_2) \\ &+ \int_{\Omega} (g_k * u_0 - u_0 - d_3)^2 (1 - u_1) u_2 \\ &+ \int_{\Omega} (g_k * u_0 - u_0 - d_4)^2 (1 - u_1)(1 - u_2), \end{aligned} \tag{9}$$

where g_k is a convolution kernel with $(k \times k)$ -size window and $\mathbf{d} = (d_1, d_2, d_3, d_4)$ is a vector-valued function. The local multiphase Chan–Vese model (local MCV) requires minimizing the energy functional

$$\mathcal{E}_{CVloc}(\mathbf{c}, \mathbf{d}, H(\Phi)) = \mathcal{E}_{CV}(\mathbf{c}, H(\Phi)) + \beta \mathcal{E}_{loc}(\mathbf{d}, H(\Phi)). \tag{10}$$

With g_k as a low-pass filter, the image difference between the filtered cartoon image and the original cartoon image will have its edges properly identified and the areas with slowly varying intensities disregarded. By incorporating the image difference, the model would take into account weak edges and illumination bias.

The effectiveness of incorporating (9) depends on the choice of the convolution kernel g_k . Since g_k needs to be a low-pass filter, one issue to be aware of is over-smoothing of the edges. For the purpose of our paper, we use a Gaussian filter, which weighs more pixels close to the center than pixels distant from it. This property effectively preserves the edges of the image as the image is being smoothed. Moreover, the choice of standard deviation for the Gaussian filter enables greater control in the amount of smoothing.

3.2 Ginzburg–Landau approximation for the local MCV model

Computing the minimizer for the energy functional (10) can be computationally expensive. For example, see the work of Getreuer [29]. Instead, one could alternatively obtain an approximate solution by threshold dynamics, proposed by Esedoglu and Tsai [24]. To this end, we approximate the energy functional (10) by a sequence of energies

$$\mathcal{E}_{CVloc}^\epsilon(\mathbf{c}, \mathbf{d}, \mathbf{u}) = \lambda \mathcal{E}_{fid}(\mathbf{c}, \mathbf{u}) + \mu \mathcal{E}_{GL}^\epsilon(\mathbf{u}) + \beta \mathcal{E}_{loc}(\mathbf{d}, \mathbf{u}), \quad (11)$$

where the perimeter term \mathcal{E}_{per} in the multiphase CV model (8) is replaced by the Ginzburg–Landau functional

$$\mathcal{E}_{GL}^\epsilon(\mathbf{u}) = \epsilon \int_{\Omega} (|\nabla u_1|^2 + |\nabla u_2|^2) + \frac{1}{\epsilon} \int_{\Omega} (W(u_1) + W(u_2)),$$

where $W(u) = u^2(1 - u)^2$. Applying the results of [49, 50], it can be shown that $\mathcal{E}_{GL}^\epsilon(\mathbf{u})$ Γ -converges to $\mathcal{E}_{per}(\mathbf{u})$ as $\epsilon \rightarrow 0^+$. Note that everything in the new energy functional is now expressed in terms of the new functions u_i instead of $H(\phi_i)$ for $i = 1, 2$.

By calculus of variations, keeping \mathbf{d} and \mathbf{u} fixed and minimizing with respect to \mathbf{c} , we obtain

$$\begin{aligned} c_1 &= \mathcal{A}[u_0, u_1, u_2] \\ c_2 &= \mathcal{A}[u_0, u_1, 1 - u_2] \\ c_3 &= \mathcal{A}[u_0, 1 - u_1, u_2] \\ c_4 &= \mathcal{A}[u_0, 1 - u_1, 1 - u_2], \end{aligned} \quad (12)$$

where

$$\mathcal{A}[u_0, u_1, u_2] = \frac{\int_{\Omega} u_0 u_1 u_2}{\int_{\Omega} u_1 u_2}.$$

Similarly, keeping \mathbf{c} and \mathbf{u} fixed and minimizing with respect to \mathbf{d} , we obtain

$$\begin{aligned} d_1 &= \mathcal{A}[g_k * u_0 - u_0, u_1, u_2] \\ d_2 &= \mathcal{A}[g_k * u_0 - u_0, u_1, 1 - u_2] \\ d_3 &= \mathcal{A}[g_k * u_0 - u_0, 1 - u_1, u_2] \\ d_4 &= \mathcal{A}[g_k * u_0 - u_0, 1 - u_1, 1 - u_2]. \end{aligned} \quad (13)$$

Keeping now \mathbf{c} and \mathbf{d} fixed and minimizing $\mathcal{E}_{CVloc}^\epsilon$ with respect to \mathbf{u} , we deduce the Euler–Lagrange equations for \mathbf{u} . It is a common technique to parametrize the descent direction by an artificial time variable $t \geq 0$ and initialize $\mathbf{u}(x, 0)$. The equations for $\mathbf{u}(x, t) = (u_1(x, t), u_2(x, t))$ are

$$\begin{aligned} \frac{\partial u_1}{\partial t} &= -\lambda \mathcal{L}(u_0, \mathbf{c}, u_2)[u_1] \\ &\quad - \beta \mathcal{L}(g_k * u_0 - u_0, \mathbf{c}, u_2)[u_1] \\ &\quad + \mu \left(2\epsilon \Delta u_1 - \frac{1}{\epsilon} W'(u_1) \right) \end{aligned} \quad (14)$$

$$\begin{aligned} \frac{\partial u_2}{\partial t} &= -\lambda \mathcal{L}(u_0, \bar{\mathbf{c}}, u_1)[u_2] \\ &\quad - \beta \mathcal{L}(g_k * u_0 - u_0, \bar{\mathbf{c}}, u_1)[u_2] \\ &\quad + \mu \left(2\epsilon \Delta u_2 - \frac{1}{\epsilon} W'(u_2) \right), \end{aligned} \quad (15)$$

where $\bar{\mathbf{c}} = (c_1, c_3, c_2, c_4)$ and the operator $\mathcal{L}(u_0, \mathbf{c}, u_2)[u_1]$ is constant with respect to u_1 , is linear in u_2 , and is given by

$$\begin{aligned} \mathcal{L}(u_0, \mathbf{c}, u_2)[u_1] &= (c_1 - u_0)^2 u_2 + (c_2 - u_0)^2 (1 - u_2) \\ &\quad - (c_3 - u_0)^2 u_2 - (c_4 - u_0)^2 (1 - u_2). \end{aligned}$$

At $t = 0$, we initialize u_1 and u_2 as the checkerboard functions

$$\begin{aligned} u_1(x, 0) &= \mathbb{1}_{\left\{ \sin \frac{\pi x_1}{3} \sin \frac{\pi x_2}{3} > 0 \right\}} \\ u_2(x, 0) &= \mathbb{1}_{\left\{ \sin \frac{\pi x_1}{10} \sin \frac{\pi x_2}{10} > 0 \right\}}, \end{aligned} \quad (16)$$

where $\mathbb{1}$ is the characteristic function, since initialization of the checkerboard function was observed to have faster convergence to the solution for the two-phase model [29].

3.3 MBO scheme for solving the local MCV

In order to solve the above system of parabolic PDEs, we implement the MBO scheme [46, 47]. The underlying idea is to solve the PDE in three steps: solve the linear ODE

$$\frac{\partial w}{\partial t} = -\lambda \mathcal{L}(u_0, \mathbf{c}, u)[w] - \beta \mathcal{L}(g_k * u_0 - u_0, \mathbf{c}, u)[w], \quad (17)$$

solve the heat equation

$$\frac{\partial v}{\partial t} = 2\mu \epsilon \Delta v, \quad (18)$$

and apply thresholding, which corresponds to solving the nonlinear ODE

$$\frac{\partial u}{\partial t} = -\frac{\mu}{\epsilon} W'(u). \quad (19)$$

We use the fact that as $\epsilon \rightarrow 0$, the solution is approaching one of the steady states ($u = 0$ and $u = 1$). Since we are dealing with a system of equations, we need to alternate the steps between the two functions u_1 and u_2 .

Using the MBO scheme, we develop an iterative algorithm that computes a sequence of solutions $\{\mathbf{c}^n, \mathbf{d}^n, \mathbf{u}^n\}$. After initialization of $\mathbf{u}^0 = (u_1^0, u_2^0)$ as the checkerboard functions (16) and the computation of

$$\tilde{u}_0 = g_k * u_0 - u_0,$$

for each iteration $n \in \mathbb{N}$, we proceed as follows:

1. Compute the average intensities

$$\begin{aligned}
 c_1^n &= \mathcal{A}[u_0, u_1^n, u_2^n] \\
 c_2^n &= \mathcal{A}[u_0, u_1^n, 1 - u_2^n] \\
 c_3^n &= \mathcal{A}[u_0, 1 - u_1^n, u_2^n] \\
 c_4^n &= \mathcal{A}[u_0, 1 - u_1^n, 1 - u_2^n]
 \end{aligned}
 \tag{20}$$

and average differences of intensities

$$\begin{aligned}
 d_1^n &= \mathcal{A}[\tilde{u}_0, u_1^n, u_2^n] \\
 d_2^n &= \mathcal{A}[\tilde{u}_0, u_1^n, 1 - u_2^n] \\
 d_3^n &= \mathcal{A}[\tilde{u}_0, 1 - u_1^n, u_2^n] \\
 d_4^n &= \mathcal{A}[\tilde{u}_0, 1 - u_1^n, 1 - u_2^n].
 \end{aligned}
 \tag{21}$$

- Let v_1^{n+1} be the solution of the ODE (17), with initial data u_1^n , computed at time dt for the operator $\mathcal{L}(\cdot, \mathbf{c}, u_2^n)$. This can be easily solved via a finite difference scheme of the form

$$\begin{aligned}
 \frac{v_1^{n+1} - u_1^n}{dt} &= -\lambda \mathcal{L}(u_0, \mathbf{c}, u_2^n)[u_1^n] \\
 &\quad - \beta \mathcal{L}(\tilde{u}_0, \mathbf{c}, u_2^n)[u_1^n].
 \end{aligned}
 \tag{22}$$

- Let w_1^{n+1} be the solution of the heat Eq. (18), with initial data v_1^{n+1} , computed at time dt . The equation can be solved in the Fourier domain as

$$\mathcal{F}_2(w_1^{n+1}) = \frac{1}{1 + 2\mu dt|\xi|^2} \mathcal{F}_2(v_1^{n+1}).
 \tag{23}$$

- Threshold to approach steady-state solutions of Eq. (19) as follows:

$$u_1^{n+1} = \begin{cases} 0 & \text{if } w_1^{n+1} \in (-\infty, \frac{1}{2}] \\ 1 & \text{if } w_1^{n+1} \in (\frac{1}{2}, +\infty). \end{cases}
 \tag{24}$$

- Set v_2^{n+1} to be the solution of the ODE (17), with initial data u_2^n , computed at time dt , for the operator $\mathcal{L}(\cdot, \bar{\mathbf{c}}, u_1^{n+1})$. This can be solved, as before, via a finite difference scheme of the form

$$\begin{aligned}
 \frac{v_2^{n+1} - u_2^n}{dt} &= -\lambda \mathcal{L}(u_0, \bar{\mathbf{c}}, u_1^{n+1})[u_2^n] \\
 &\quad - \beta \mathcal{L}(\tilde{u}_0, \bar{\mathbf{c}}, u_1^{n+1})[u_2^n].
 \end{aligned}
 \tag{25}$$

- Let w_2^{n+1} be the solution of the heat Eq. (18), with initial data v_2^{n+1} , computed at time dt . As before, we solve the equation in the Fourier domain as

$$\mathcal{F}_2(w_2^{n+1}) = \frac{1}{1 + 2\mu dt|\xi|^2} \mathcal{F}_2(v_2^{n+1}).
 \tag{26}$$

- Threshold to approach steady-state solutions of Eq. (19) as follows:

$$u_2^{n+1} = \begin{cases} 0 & \text{if } w_2^{n+1} \in (-\infty, \frac{1}{2}] \\ 1 & \text{if } w_2^{n+1} \in (\frac{1}{2}, +\infty). \end{cases}
 \tag{27}$$

In order to obtain the segmented result \tilde{u} , we multiply u_2 by two and add it to u_1 to form at most four segmented regions. If we simply add u_1 and u_2 together, we would have at most three regions. The algorithm for minimizing (10) is summarized in Algorithm 2. Results are shown and discussed in Sect. 5.

Algorithm 2: MBO scheme for local MCV

Input: Image u_0 , parameters λ, μ, β, dt

1: Compute

$$\tilde{u}_0 = g_k * u_0 - u_0$$

using the Gaussian filter, preferably `imgaussfilt` in MATLAB.

2: Initialize u_1 and u_2 as in (16).

3: **for** $i = 1$ to n **do**

4: Compute the average intensities \mathbf{c}^i as in (20).

5: Compute the average intensities \mathbf{d}^i as in (21).

6: Compute the solution \mathbf{u}^i using equations

(22)-(27).

7: **end for**

8: Combine u_1^n and u_2^n to obtain multiphase image \tilde{u}^n ,

i.e.

$$\tilde{u}^n = u_1^n + 2u_2^n$$

Output: Segmented Image \tilde{u}^n

4 Texture segmentation using empirical wavelet transform

STM images may be comprised of various texture patterns, as shown in Figs. 3, 4. These patterns are not in general simple waves but rather a combination of several simple oscillations. The goal of this section is to partition the image textures into several components, each grouping together pixels belonging to a similar pattern. Texture segmentation is in general a difficult task because it needs to take into account all types of variability within textures. Such difficulties can be leveraged in the case of particular textures, as for STM images, since the textures correspond to periodic patterns at different frequencies and with different orientations.

Directional image decomposition methods involve a decomposition of the Fourier spectrum into basis elements. These methods include Gabor filters [21–23, 39, 68], wavelets [2, 60, 64], curvelets [3, 12, 13, 58], and shearlets [42]. However, they are not adaptive and may result in incoherent partitions for STM-type images. Adaptive methods, on the other hand, provide better image decomposition since the basis elements are generated by the information contained in the image itself. Among these adaptive methods, the 2D

variational mode decomposition (VMD) is a non-recursive, fully adaptive algorithm that sparsely decomposes signals/images into ensembles of constituent modes [19, 20, 73]. By minimizing an energy functional, this method simultaneously retrieves a given number of modes (texture patterns) together with their supports and the frequencies around which they are band-limited. To accommodate microscopy images, where texture patterns are combinations of simple modes, the energy functional is adapted to lattices by coupling several modes with a single support function. The solution to the energy functional is optimized using alternating direction method of multipliers (ADMM) [26, 33] and the MBO scheme, requiring several parameters to tune for the fidelity, penalty, and regularization terms involved as well as the convergence rate of ADMM [7]. The VMD is an effective segmentation model, which has been demonstrated to work well on a broad variety of images. However, for our purpose, it presents two major inconveniences: (1) the large number of parameters to be tuned, which can become time-consuming when trying to achieve the best texture segmentation result, and (2) the explicit number of active modes required in the decomposition, which is restrictive.

An alternative method is the empirical wavelet transform (EWT) [30]. It is very well adapted to patterns specific to STM image textures, and it automatically finds the number of modes while requiring very few parameters to tune. Using EWT, we propose a texture segmentation algorithm adapted to microscopy image textures, which consists of three steps: (1) perform the EWT on the texture component, (2) construct a feature matrix based on the processed EWT coefficients, and (3) apply a clustering algorithm to the feature matrix in order to obtain the final segmentation. As detailed below, the main contributions in the EWT algorithm are the improvement in boundary detection and partitioning the Fourier spectrum and the selection of texture features based on their local energy.

4.1 The empirical wavelet transform

The EWT was originally proposed as a signal decomposition method that detects and separates the signals' principal harmonic modes. The principal modes are modeled as amplitude-modulated–frequency-modulated (AM-FM) signals with compact support in the Fourier domain [30]. The EWT consists of two steps: (1) it partitions the Fourier spectrum into N supports and it builds in the Fourier domain a filter bank, where each wavelet filter corresponds to a support, and (2) it filters the input signal with the obtained filter bank to produce the different components. The filter bank consists of N wavelet filters: one low-pass filter corresponding to the approximation component and $N - 1$ band-pass filters corresponding to the detailed components.

The partitioning of the Fourier spectrum is as important as building the adaptive wavelets since it provides information about the principal harmonic components. Several approaches to perform the boundaries detection in the Fourier domain were investigated by [30–32]. In particular, Gilles and Healy proposed a fully automatic algorithm based on a combination of a scale-space representation and Otsu's method [31]. The advantage of this approach is in that it automatically finds the number N of expected modes and detects the boundaries of the Fourier supports. We used this approach in all experiments.

Assume that the Fourier spectrum is partitioned into N contiguous segments with boundaries $\{\omega_n\}_{n=0}^N$, where $\omega_0 = 0$ and $\omega_N = \pi$ (see Fig. 6). Then, based on Meyer's wavelet formulation, we construct a filter bank of wavelets

$$\{\phi_1(x), \{\psi_n(x)\}_{n=1}^{N-1}\},$$

on the corresponding segments $[\omega_{n-1}, \omega_n]$. The Fourier transform (denoted as \mathcal{F}_1 in the 1D case) of the empirical scaling function is given by

$$\mathcal{F}_1(\phi_1)(\omega) = \begin{cases} 1 & \text{if } |\omega| \leq (1 - \gamma)\omega_1 \\ \cos \left[\frac{\pi}{2} \mathcal{B} \left(\frac{1}{2\gamma\omega_1} (|\omega| - (1 - \gamma)\omega_1) \right) \right] & \text{if } (1 - \gamma)\omega_1 \leq |\omega| \leq (1 + \gamma)\omega_1 \\ 0 & \text{if otherwise} \end{cases}, \quad (28)$$

whereas the Fourier transforms of the empirical wavelets are given by

$$\mathcal{F}_1(\psi_n)(\omega) = \begin{cases} 1 & \text{if } (1 + \gamma)\omega_n \leq |\omega| \leq (1 - \gamma)\omega_{n+1} \\ \cos \left[\frac{\pi}{2} \mathcal{B} \left(\frac{1}{2\gamma\omega_{n+1}} (|\omega| - (1 - \gamma)\omega_{n+1}) \right) \right] & \text{if } (1 - \gamma)\omega_{n+1} \leq |\omega| \leq (1 + \gamma)\omega_{n+1} \\ \sin \left[\frac{\pi}{2} \mathcal{B} \left(\frac{1}{2\gamma\omega_n} (|\omega| - (1 - \gamma)\omega_n) \right) \right] & \text{if } (1 - \gamma)\omega_n \leq |\omega| \leq (1 + \gamma)\omega_n \\ 0 & \text{if otherwise} \end{cases}, \quad (29)$$

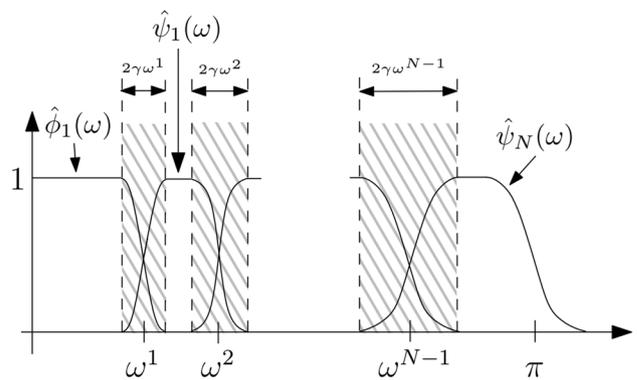


Fig. 6 Partitioning of the Fourier spectrum of a 1D signal

for $n = 1, \dots, N$. The function \mathcal{B} is an arbitrary $C^k([0, 1])$ function satisfying the properties that $\mathcal{B}(t) = 0$ if $t \leq 0$, $\mathcal{B}(t) = 1$ if $t \geq 1$, and $\mathcal{B}(t) + \mathcal{B}(1 - t) = 1$ and $\mathcal{B}(t) \in (0, 1)$ for all $t \in [0, 1]$. The parameter γ is chosen to ensure that two consecutive transition areas (shown as dashed areas in Fig. 6) do not overlap. As shown in [30], a proper selection of γ guarantees that the filter bank $\{\phi_1, \{\psi_n\}_{n=1}^{N-1}\}$ is a tight frame in $L^2(\mathbb{R})$. Then, the EWT is defined in the same way as the classical wavelet transform. For the signal function f , the detailed coefficients are given as

$$\mathcal{W}_f^\mathcal{E}(n, x) = \mathcal{F}_1^* \left(\mathcal{F}_1(f)(\omega) \overline{\mathcal{F}_1(\psi_n)(\omega)} \right) (x), \tag{30}$$

and the approximation coefficient as

$$\mathcal{W}_f^\mathcal{E}(0, x) = \mathcal{F}_1^* \left(\mathcal{F}_1(f)(\omega) \overline{\mathcal{F}_1(\phi_1)(\omega)} \right) (x), \tag{31}$$

where \mathcal{F}_1^* stands for the inverse 1D Fourier transform.

The EWT was later generalized to 2D images for various kinds of wavelet transform, specifically tensor wavelets, Littlewood–Paley wavelet transform, the ridgelet transform, and the curvelet transform [32].

4.2 The empirical curvelet transform

Textures in STM images can be seen as oscillatory patterns with multiple orientations. Among all of the above-mentioned variants, curvelets are wavelets that take into account various orientations [13]. Therefore, its empirical counterpart, the empirical curvelet transform (ECT), is the most appropriate adaptation of EWT to partition texture images. Similar to the EWT, the ECT builds a filter bank in the Fourier domain where each filter has its support on a polar wedge.

As shown in Fig. 7b, the Fourier domain is partitioned in this case into a disk centered at the origin, which contains the low frequencies, and concentric annuli and angular sectors (polar wedges), which contain high frequencies. In order to build the filter bank adaptively, the ECT needs to detect the boundaries of the polar wedges empirically, which correspond to finding the scales for the angle and the radii. Following [32], this step can be achieved by considering the pseudo-polar Fourier transform [5, 6] and performing the previously described 1D detection to 1D spectra corresponding to averaging with respect to the frequency magnitude and orientation, respectively. Gilles et al. [32] proposed three different cases: (1) scales and angles are detected independently, (2) scales are detected first and angles are detected per each scale, and (3) angles are detected first and scales are detected per each angular sector. In this work, we will consider the second option because STM images contain texture patterns with different main orientations having varying frequency magnitudes.

In the following, we will denote $\omega = (\omega_{x_1}, \omega_{x_2})$ the frequency coordinates in the Fourier plane, $|\omega|$ its magnitude and θ its angle. Otsu’s boundary detection method will then provide N_θ number of angles and N_s^m number of scales per each angular sector $m = 1, \dots, N_\theta$. This is equivalent to obtaining the set of angular boundaries $\{\theta_m\}_{m=1}^{N_\theta}$ and the set of scale boundaries $\{\omega_n^m\}_{n=1}^{N_s^m}$ per angular sector for $m = 1, \dots, N_\theta$. Note that $\omega_1^1 = \dots = \omega_1^{N_\theta}$ because altogether they form the disk centered at the origin of the Fourier domain. The corresponding curvelet filters are then defined in the Fourier domain (we denote \mathcal{F}_2 the 2D Fourier transform) in the following way. The purely radial low-pass filter ϕ_1 is given by

$$\mathcal{F}_2(\phi_1)(\omega, \theta) = \begin{cases} 1 & \text{if } |\omega| \leq (1 - \gamma)\omega_1, \\ \cos \left[\frac{\pi}{2} \mathcal{B} \left(\frac{1}{2\gamma\omega_1} (|\omega| - (1 - \gamma)\omega_1) \right) \right] & \text{if } (1 - \gamma)\omega_1 \leq |\omega| \leq (1 + \gamma)\omega_1, \\ 0 & \text{if otherwise,} \end{cases} \tag{32}$$

where $\omega_1 = \omega_1^1 = \dots = \omega_1^{N_\theta}$. The polar curvelet associated with the polar wedge $I_n^m := [\theta_m, \theta_{m+1}] \times [\omega_n^m, \omega_{n+1}^m]$ can be written as

$$\mathcal{F}_2(\psi_{m,n})(\omega, \theta) = W_n^m(|\omega|) V_m(\theta), \tag{33}$$

where the radial window W_n^m is

$$W_n^m(|\omega|) = \begin{cases} 1 & \text{if } (1 + \gamma)\omega_n^m \leq |\omega| \leq (1 - \gamma)\omega_{n+1}^m, \\ \cos \left[\frac{\pi}{2} \mathcal{B} \left(\frac{1}{2\gamma\omega_{n+1}^m} (|\omega| - (1 - \gamma)\omega_{n+1}^m) \right) \right] & \text{if } (1 - \gamma)\omega_{n+1}^m \leq |\omega| \leq (1 + \gamma)\omega_{n+1}^m, \\ \sin \left[\frac{\pi}{2} \mathcal{B} \left(\frac{1}{2\gamma\omega_n^m} (|\omega| - (1 - \gamma)\omega_n^m) \right) \right] & \text{if } (1 - \gamma)\omega_n^m \leq |\omega| \leq (1 + \gamma)\omega_n^m, \\ 0 & \text{if otherwise,} \end{cases} \tag{34}$$

for $n \neq N_s^m - 1$ and

$$W_{N_s^m-1}^m(|\omega|) = \begin{cases} 1 & \text{if } (1 + \gamma)\omega_{N_s^m-1}^m \leq |\omega|, \\ \sin \left[\frac{\pi}{2} \mathcal{B} \left(\frac{1}{2\gamma\omega_{N_s^m-1}^m} (|\omega| - (1 - \gamma)\omega_{N_s^m-1}^m) \right) \right] & \text{if } (1 - \gamma)\omega_{N_s^m-1}^m \leq |\omega| \leq (1 + \gamma)\omega_{N_s^m-1}^m, \\ 0 & \text{if otherwise,} \end{cases} \tag{35}$$

for $n = N_s^m - 1$, while the angular window V_m is

$$V_m(\theta) = \begin{cases} 1 & \text{if } \theta_m + \Delta\theta \leq \theta \leq \theta_{m+1} - \Delta\theta, \\ \cos \left[\frac{\pi}{2} \mathcal{B} \left(\frac{1}{2\Delta\theta} (\theta - \theta_{m+1} + \Delta\theta) \right) \right] & \text{if } \theta_{m+1} - \Delta\theta \leq \theta \leq \theta_{m+1} + \Delta\theta, \\ \sin \left[\frac{\pi}{2} \mathcal{B} \left(\frac{1}{2\Delta\theta} (\theta - \theta_m + \Delta\theta) \right) \right] & \text{if } \theta_m - \Delta\theta \leq \theta \leq \theta_m + \Delta\theta, \\ 0 & \text{if otherwise.} \end{cases} \tag{36}$$

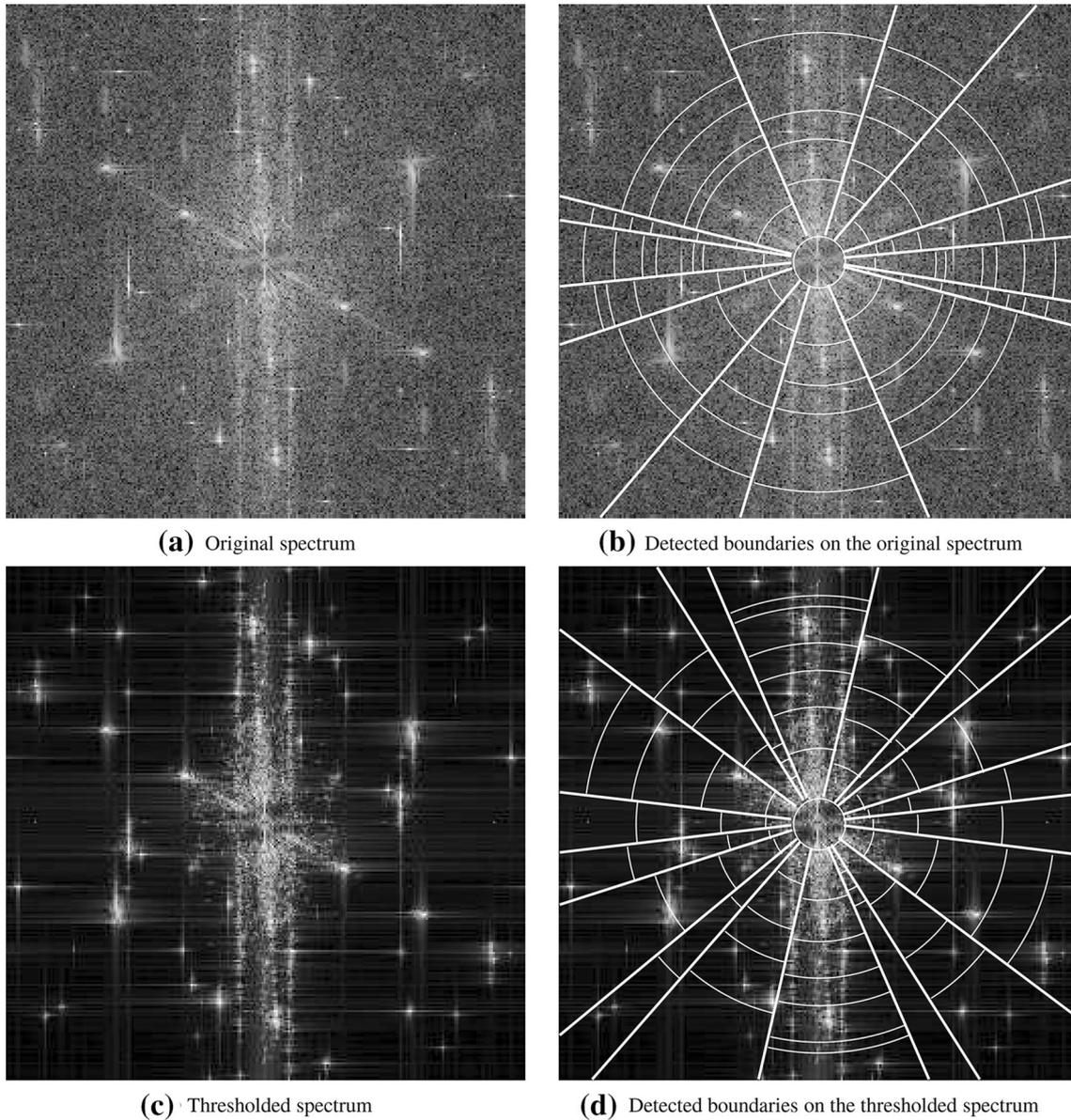


Fig. 7 Comparison of partitions obtained from the original spectrum and its thresholded version. The spectrum shown is reproduced and modified from [35] with permission. STM image spectrum is copyright American Chemical Society

The parameters γ and $\Delta\theta$ are chosen in order to guarantee the tight frame property (see [32] for details). This leads to the construction of the filter bank of empirical curvelets

$$\left\{ \begin{array}{l} \phi_1(x), \{\psi_{m,n}(x)\} \\ m = 1, \dots, N_\theta \\ n = 1, \dots, N_s^m - 1 \end{array} \right\}. \quad (37)$$

From (37), the empirical curvelet transform of the texture component v is given by the detailed coefficients as

$$\mathcal{W}_v^{\mathcal{EC}}(m, n, x) = \mathcal{F}_2^* \left(\mathcal{F}_2(v) \overline{\mathcal{F}_2(\psi_{m,n})} \right)(x), \quad (38)$$

and the approximation coefficients as

$$\mathcal{W}_v^{\mathcal{EC}}(0, 0, x) = \mathcal{F}_2^* \left(\mathcal{F}_2(v) \overline{\mathcal{F}_2(\phi_1)} \right)(x), \quad (39)$$

where \mathcal{F}_2^* stands for the inverse 2D Fourier transform. We can reconstruct the image $v(x)$ by the inverse transform:

$$\begin{aligned} v(x) = & \mathcal{F}_2^* \left(\mathcal{F}_2(\mathcal{W}_v^{\mathcal{EC}}(0, 0, \cdot)) \overline{\mathcal{F}_2(\phi_1)} \right) \\ & + \sum_{m=1}^{N_\theta} \sum_{n=1}^{N_s^m - 1} \mathcal{F}_2^* \left(\mathcal{W}_v^{\mathcal{EC}}(m, n, \cdot) \overline{\mathcal{F}_2(\psi_{m,n})} \right)(x). \end{aligned} \quad (40)$$

Each empirical curvelet subband $\mathcal{W}_v^{\mathcal{EC}}(m, n, \cdot)$ contains either some textural pattern or noise. In many cases, some of these curvelets may have extremely weak oscillatory patterns or pure noise, deeming them useless in our texture analysis. Therefore, we propose a preprocessing step and a post-processing step in order to obtain only empirical curvelets with meaningful information.

4.3 Improved boundaries detection

In this section, we propose modifications in the boundary detection algorithm in order to reduce the number of irrelevant polar wedges. Since the detection is performed in the pseudo-polar Fourier domain, the following processing will also be performed in the pseudo-polar domain. Hereafter, we will denote $\mathcal{F}_p(v)$ as the pseudo-polar Fourier transform of the input texture image v .

Our first improvement is a preprocessing step before the actual detection. Since the goal is to isolate clusters of high-magnitude frequencies in $\mathcal{F}_p(v)$, we suggest thresholding the Fourier coefficients in order to remove all frequencies that are not relevant. Let T be the hard thresholding operator defined as

$$T(a, \tau) = \begin{cases} 0 & \text{if } |a| \leq \tau \\ a & \text{if } |a| > \tau \end{cases} \quad (41)$$

The preprocessing step consists of performing the detection on $T(\mathcal{F}_p(v), \tau)$ instead of $\mathcal{F}_p(v)$. The threshold τ is chosen as a certain percentile of the magnitude of the Fourier coefficients: We create a vector whose entries are the magnitudes of the Fourier coefficients listed in increasing order and we set τ to be the value of the $(n \times p)$ th entry of the vector, where n is the length of the vector and p the specified percentile (*i.e.*, $p \in (0, 1)$). Figure 7 illustrates the effects of

performing such thresholding. Figure 7a, c shows the original spectrum as well as its thresholded version (using a 92 percentile). Figure 7b, d provides the corresponding detected partitions. We note that the partition obtained from the thresholded spectrum provides a separate angular sector for the high-magnitude clusters near the top right corner, while these clusters are associated with another angular sector in the partition obtained from the original spectrum.

Unfortunately, sometimes some meaningless polar wedges are detected in each angular sector. Our second improvement aims at removing these useless polar wedges by merging them with their neighbors. Suppose that $\{\theta_m\}_{m=1}^{N_\theta}$ and $\{\omega_n^m\}_{n=1}^{N_s^m}\}_{m=1}^{N_\theta}$ are the set of angles and scales detected on $T(\mathcal{F}_p(v), \tau)$. In the pseudo-polar domain, a polar wedge corresponds to the rectangle I_n^m , whose area we denote by $A_n^m = \text{area}(I_n^m) = (\omega_{n+1}^m - \omega_n^m)(\theta_{m+1} - \theta_m)$. We define the information density per polar wedge by

$$M_n^m = \frac{\|T(\mathcal{F}_p(v), \tau)\mathbb{1}_{I_n^m}\|_1}{A_n^m}, \quad (42)$$

where $\mathbb{1}_{I_n^m}$ is the characteristic function over the domain I_n^m . As a reference, we use the largest density $\tilde{M} = \max_{m,n} M_n^m$ and we threshold at a certain fraction $\eta = 0.1$ as follows. For a given angular sector, we start from the polar wedge that is the farthest from the origin until we reach the one closer to the origin. If $M_n^m \leq \eta\tilde{M}$, then the I_n^m polar wedge is irrelevant and we merge it with I_{n-1}^m ; that is, we remove ω_{n-1}^m from the list $\{\omega_n^m\}_{n=1}^{N_s^m}$ and update $N_s^m := N_s^m - 1$. If $M_n^m > \eta\tilde{M}$, we move to the next polar wedge I_{n-1}^m and repeat the procedure. The corresponding merging algorithm is summarized in Algorithm 3.

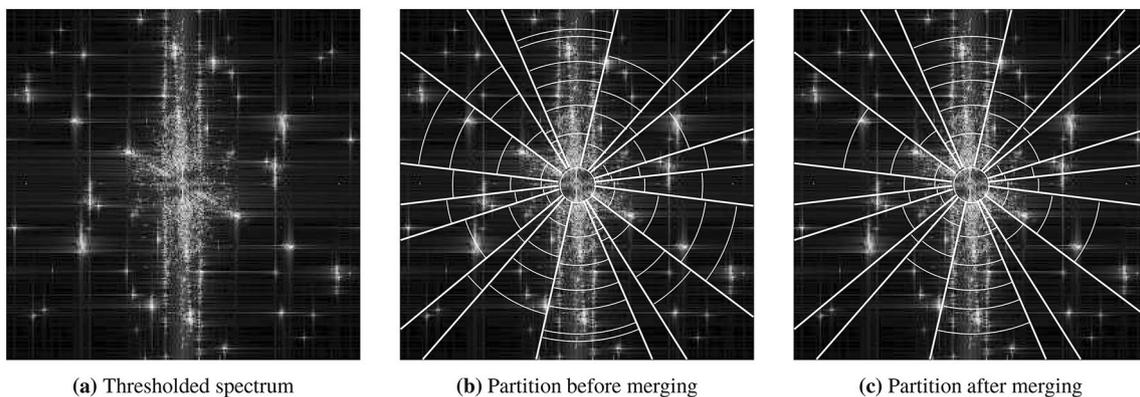


Fig. 8 Influence of the merging algorithm on the obtained partitions. The spectrum shown is reproduced and modified from [35] with permission. STM image spectrum is copyright American Chemical Society

Algorithm 3: Merging Curvelet partition

Input: Thresholded Fourier domain $T(\mathcal{F}_P(v), \tau)$, original boundaries $\{\theta_m\}_{m=1}^{N_\theta}$ and $\left\{\{\omega_n^m\}_{n=1}^{N_s^m}\right\}_{m=1}^{N_\theta}$

- 1: Compute M_n^m for $m = 1 \dots N_\theta$ and $n = 1 \dots N_s^m$ according to (42).
- 2: Compute $\tilde{M} = \max_{m,n} M_n^m$.
- 3: **for** $i = 1$ to N_θ **do**
- 4: Set $j := N_s^i$.
- 5: **while** $j \geq 3$ **do**
- 6: **if** $M_{j-1}^i < 0.10\tilde{M}$ **then**
- 7: Remove ω_{j-1}^i from $\left\{\{\omega_n^m\}_{n=1}^{N_s^m}\right\}_{m=1}^{N_\theta}$.
- 8: Set $N_s^i := N_s^i - 1$.
- 9: **end if**
- 10: $j := j - 1$.
- 11: **end while**
- 12: **end for**

Output: Updated boundaries $\left\{\{\theta_m\}_{m=1}^{N_\theta}, \left\{\{\omega_n^m\}_{n=1}^{N_s^m}\right\}_{m=1}^{N_\theta}\right\}$.

Partitions obtained before and after merging are illustrated in Fig. 8. It is easy to see that some polar wedges having less information were merged to form a new set of polar wedges having a minimum amount of useful information. (See, for instance, the most vertical angular sector, the thin outer wedge, no longer exists in the updated partition).

Based on the updated partition, the empirical curvelet filter bank is then constructed accordingly to (32) and (33).

The fully modified empirical curvelet transform is summarized in Algorithm 4.

As shown in [32], the tight frame property depends only on the construction process of the curvelet filters themselves and does not depend on how the support detection is done. Therefore, the original proof remains valid even within the framework of our modified support detection algorithm, implying that the tight frame property is preserved.

Algorithm 4: Modified Empirical Curvelet Transform

Input: Image $v(x)$, Threshold Value τ

- 1: Compute the Pseudo-Polar FFT $\mathcal{F}_P(v)$.
- 2: Threshold the Fourier coefficients to obtain $T(\mathcal{F}_P(v), \tau)$.
- 3: Detect the original partition $\left\{\{\theta_m\}_{m=1}^{N_\theta}, \left\{\{\omega_n^m\}_{n=1}^{N_s^m}\right\}_{m=1}^{N_\theta}\right\}$ using Otsu's Method as described in [31, 32].
- 4: Compute the updated set $\left\{\{\theta_m\}_{m=1}^{N_\theta}, \left\{\{\omega_n^m\}_{n=1}^{N_s^m}\right\}_{m=1}^{N_\theta}\right\}$ by applying the merging algorithm Algorithm 3.
- 5: Construct the corresponding curvelet filter bank $\mathcal{B}^{\mathcal{M}\mathcal{E}\mathcal{C}} = \left\{\phi_1(x), \{\psi_{m,n}(x)\}_{\substack{m=1,\dots,N_\theta \\ n=1,\dots,N_s^m-1}}\right\}$ accordingly to (32)-(33).
- 6: Filter $v(x)$ using (38)-(39) to obtain $\mathcal{W}_v^{\mathcal{M}\mathcal{E}\mathcal{C}} = \left\{\mathcal{W}_v^{\mathcal{M}\mathcal{E}\mathcal{C}}(0, 0, x), \{\mathcal{W}_v^{\mathcal{M}\mathcal{E}\mathcal{C}}(m, n, x)\}_{\substack{m=1,\dots,N_\theta \\ n=1,\dots,N_s^m-1}}\right\}$

Output: Spectrum boundaries $\left\{\{\theta_m\}_{m=1}^{N_\theta}, \left\{\{\omega_n^m\}_{n=1}^{N_s^m}\right\}_{m=1}^{N_\theta}\right\}$, empirical curvelet filter bank $\mathcal{B}^{\mathcal{M}\mathcal{E}\mathcal{C}}$, empirical curvelets coefficients $\mathcal{W}_v^{\mathcal{M}\mathcal{E}\mathcal{C}}$.

4.4 Texture features

After applying our modified empirical curvelet transform to the texture component of the image, we finally construct the relevant information to characterize different textures. This information could be directly given by the empirical curvelet coefficients by reshaping each one into a vector corresponding to each pixel. However, this kind of feature vector does not have any inherent spatial information of the local neighborhood of its corresponding pixel. Instead, we consider computing the local “energy” of the curvelet coefficients. Then, we build a set of feature vectors associated with each pixel of the texture component. This set will form together a feature matrix whereby a clustering algorithm can be applied to it.

Define the local energy at (m, n, x) as

$$E_{m,n}^{\mathcal{M}\mathcal{E}\mathcal{C}}(x) = \frac{\|\mathbb{1}_{B(x,r_{m,n})} \mathcal{W}_v^{\mathcal{M}\mathcal{E}\mathcal{C}}(m, n, \cdot)\|_2}{|B(x, r_{m,n})|} \tag{43}$$

where $B(x, r_{m,n})$ is a $(2r_{m,n} + 1) \times (2r_{m,n} + 1)$ neighborhood window centered at x . The radius $r_{m,n}$ is determined by the frequency of the texture pattern so that $E_{m,n}^{\mathcal{M}\mathcal{E}\mathcal{C}}(x)$ captures enough information around x . Hence, with ω_n^m related to frequency, we choose $r_{m,n}$ proportional to $\lceil \frac{1}{\omega_n^m} \rceil$ so that the radius relates to the period of the texture pattern. In practice, we set $r_{m,n} = \lceil \frac{\pi}{\omega_n^m} \rceil$. Note that for pixels near or at the border of the image, we apply symmetric padding to align with our assumption that texture patterns are periodic.

After calculating (43) for each pixel for each empirical curvelet subband, we construct the feature matrix \mathcal{D} by casting all energy matrices $E_{m,n}^{\mathcal{M}\mathcal{E}\mathcal{C}}$ as vectors so that they form the columns of \mathcal{D} . Thus, \mathcal{D} has the form

$$\mathcal{D} = \begin{pmatrix} | & \dots & | & | & \dots & | \\ E_{1,1}^{\mathcal{M}\mathcal{E}\mathcal{C}} & \dots & E_{1,N_s^1-1}^{\mathcal{M}\mathcal{E}\mathcal{C}} & E_{2,1}^{\mathcal{M}\mathcal{E}\mathcal{C}} & \dots & E_{N_\theta, N_s^{N_\theta}-1}^{\mathcal{M}\mathcal{E}\mathcal{C}} \\ | & \dots & | & | & \dots & | \end{pmatrix}. \tag{44}$$

Because texture patterns are locally periodic in an image, pixels belonging to the same texture pattern should both have similar energies as defined by (43). Hence, we could group the column vectors as belonging to the same class. We can then apply any clustering algorithm, such as k -means or multiclass MBO clustering [27, 45], on \mathcal{D} to identify the texture patterns of the image.

5 Experimental results

In this section, we present results of our framework applied to the images in Figs. 3, 4. All algorithms in this paper were implemented in MATLAB R2018b. The codes and the results are available at <https://github.com/kbui1993/Microscopy-Codes>.

In order to segment the images according to intensities or texture patterns, we first perform the cartoon + texture decomposition on each of them to obtain their cartoon and texture components. In our experiments, we select $\sigma = 3$ in Algorithm 1. This parameter leads to more appealing segmentation results compared to other values of σ . Moreover, $\sigma = 3$ is the minimum value for which humans could perceive region as textures [8].

After obtaining the cartoon and texture components for each image, we apply our proposed methods to each component. Our results are divided into two subsections. The cartoon segmentation results are given in Sect. 5.1, while the texture segmentation results are given in Sect. 5.2.

5.1 Cartoon segmentation results

For comparison, we apply multilevel threshold by Otsu’s method [55], k -means clustering [4], the multiphase algorithm based on the MBO scheme [24], and the proposed local multiphase segmentation algorithm (Algorithm 2) to the cartoon component of each image in Figs. 3, 4. Convergence for either multiphase methods is reached when both phase fields u_1 and u_2 do not change within the same iteration. Otherwise, the maximum number of iterations is set at 200. For each image, the values for λ and μ are set the same between both methods, but the time step dt might be different because both methods are minimizing different energy functionals. The number of clusters or segmentations are four. Finally, the standard deviation for the Gaussian filter in Algorithm 2 is set to 10.

Here, we discuss the parameter selection for both the multiphase and local multiphase methods. In our experiments, the values of λ in (8) are set to 5, 7, or 10. If there are multiple regions or regions of varying sizes in the image, the value μ should be small. Otherwise, it should be large. We adopt four values for μ : $10^{-4} \times (N \times M)$, $10^{-3} \times (N \times M)$, $10^{-2} \times (N \times M)$, and $10^{-1} \times (N \times M)$, where $(N \times M)$ is the dimension of the image. The dimensions of the images in Figs. 3, 4 are 256^2 . The parameter β from (10) is set as the same value as λ if there is intensity homogeneity. Otherwise, if there is intensity inhomogeneity, β is set to be larger than λ to ensure that potential regions with weak edges are detected. To ensure Γ -convergence of (11) when running Algorithm 2, the time step dt needs to be small (we set here

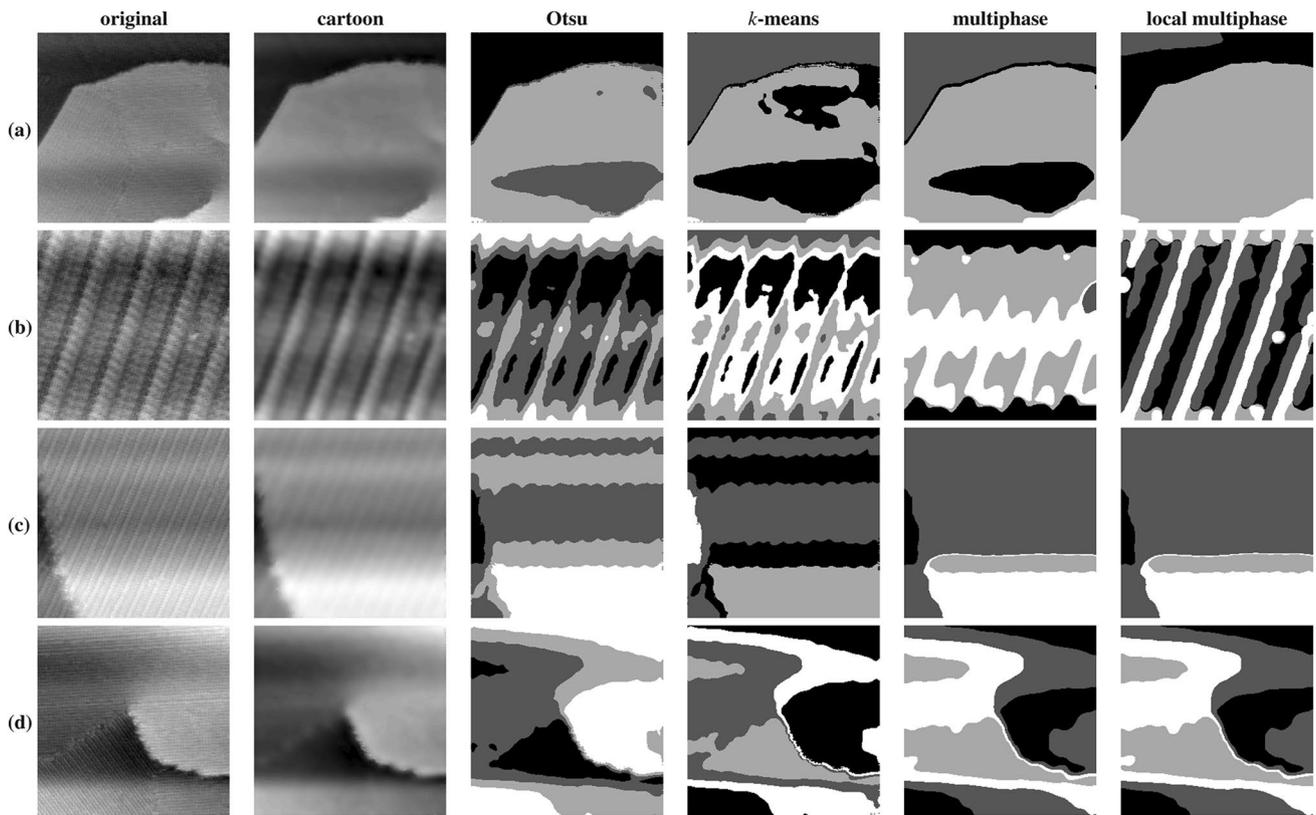


Fig. 9 Comparison between four segmentation methods. Otsu’s method and *k*-means are nonparametric. The parameters for multiphase and local multiphase are: **a** $\lambda = 10$, $\mu = 10^{-3} \times 256^2$, $\beta = 10$, $dt_{CV} = 0.75$, $dt_{CVloc} = 3.2$; **b** $\lambda = 10$, $\mu = 10^{-3} \times 256^2$, $\beta = 300$, $dt_{CV} = 4$, $dt_{CVloc} = 4$; **c** $\lambda = 10$, $\mu = 10^{-3} \times 256^2$, $\beta = 60$,

$dt_{CV} = 2$, $dt_{CVloc} = 2$; **d** $\lambda = 10$, $\mu = 10^{-3} \times 256^2$, $\beta = 10$, $dt_{CV} = 2.5$, $dt_{CVloc} = 2$. Raw scanning tunneling microscope images of cyanide on Au{111}, reproduced from [35] with permission. Images copyright American Chemical Society

$dt < 20$), and the number of iterations needs to be large, which is why the maximum is set to 200.

Because ground truth is unavailable, we cannot evaluate the segmentation accuracy, so we use an entropy-based evaluation criteria [70]. The entropy of region R_j for $j = 1, \dots, 4$ is

$$H(R_j) = - \sum_{m \in V_j} \frac{L_j(m)}{S_j} \log \frac{L_j(m)}{S_j},$$

where V_j is the set of all possible gray-level values in region R_j , $L_j(m)$ is the number of pixels in region R_j that have gray level m , and $S_j = |R_j|$ is the number of pixels in region R_j . The expected region entropy of segmentation image I is

$$H_r(I) = \sum_{j=1}^C \left(\frac{S_j}{S_I} \right) H(R_j),$$

where C is the number of segmentations (for our case, $C = 4$) and S_I is the total number of pixels in segmentation image I . The layout entropy is

$$H_l(I) = - \sum_{j=1}^C \log \frac{S_j}{S_I}.$$

The entropy-based evaluation function E is

$$E = H_r(I) + H_l(I).$$

The expected region entropy $H_r(I)$ measures the uniformity within the regions of I . If the region is very uniform, then $H_r(I)$ will be small. The layout entropy penalizes over-segmentation when the region entropy becomes small [70, 71]. Therefore, the entropy-based evaluation function E balances the trade-off between the uniformity of the individual regions with the complexity of the segmentation. Note that the smaller E is the better the segmentation.

The results of each method are shown along with the original and the cartoon images in Figs. 9, 10, 11, and 12. Additionally, the entropy values for the results are listed in Table 1. We observe that the results differ between the four methods.

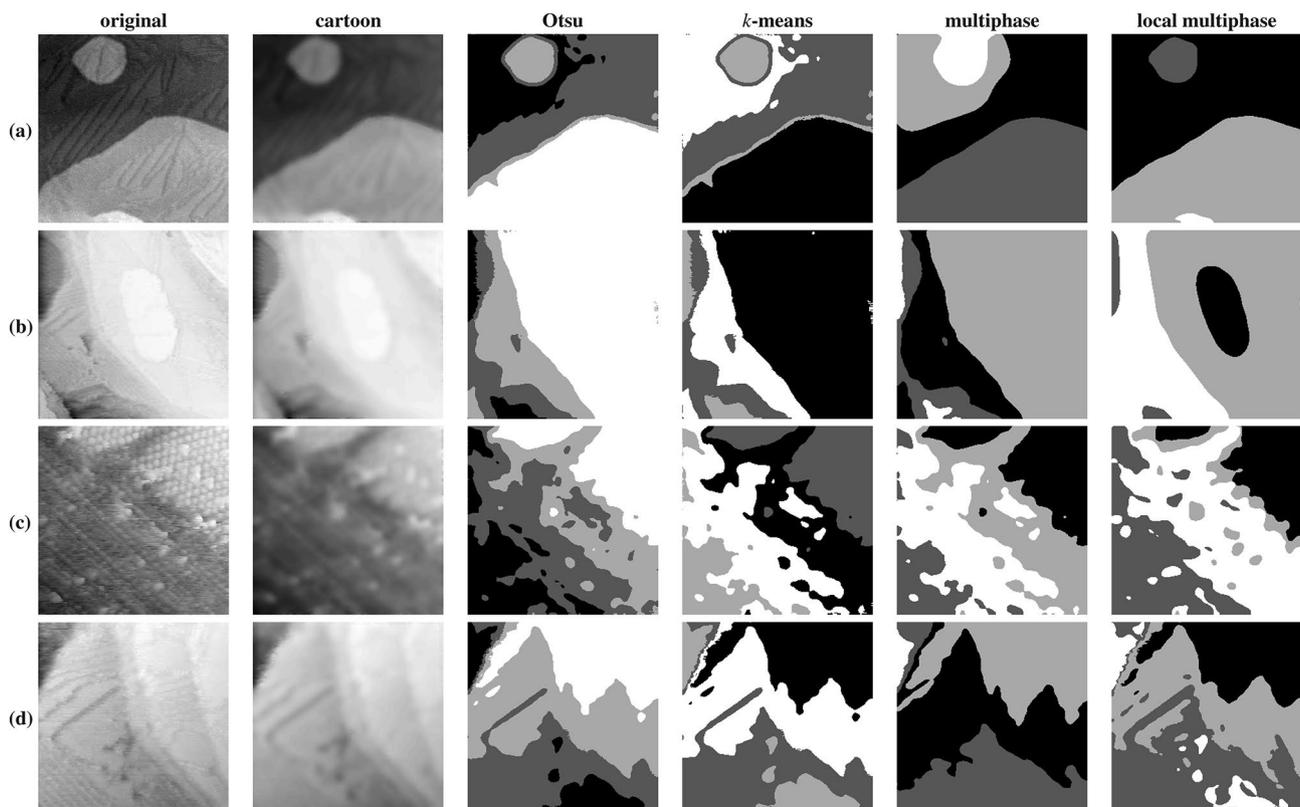


Fig. 10 Comparison between four segmentation methods. Otsu’s method and *k*-means are nonparametric. The parameters for multiphase and local multiphase are: **a** $\lambda = 5$, $\mu = 10^{-1} \times 256^2$, $\beta = 70$, $dt_{CV} = 0.35$, $dt_{CVloc} = 0.10$; **b** $\lambda = 10$, $\mu = 10^{-2} \times 256^2$, $\beta = 30$, $dt_{CV} = 0.1$, $dt_{CVloc} = 6.5$; **c** $\lambda = 7$, $\mu = 10^{-4} \times 256^2$, $\beta = 65$,

$dt_{CV} = 5$, $dt_{CVloc} = 18$; **d** $\lambda = 5$, $\mu = 10^{-4} \times 256^2$, $\beta = 50$, $dt_{CV} = 12$, $dt_{CVloc} = 0.6$. Raw scanning tunneling microscope images of cyanide on Au{111}, reproduced from [35] with permission. Images copyright American Chemical Society

In Fig. 9a, Otsu’s method, *k*-means, and multiphase detect streaks of shadow in the middle of the image along with an edge as a region. However, these should not appear as distinct regions at all. On the other hand, local multiphase performs properly: It detects smaller and more subtle regions of the cartoon image, which are located in the top left and bottom right.

Figure 9b is heavily affected by illumination bias. Hence, we have poor segmentation results from Otsu’s method, *k*-means, and multiphase. On the other hand, the result from local multiphase is able to identify the natural regions of its corresponding cartoon image.

In Fig. 9c, both multiphase and local multiphase produce results similar to the cartoon image, whereas Otsu’s method and *k*-means do not. The results from Otsu’s method and *k*-means seem sensitive to intensity inhomogeneities and shadows that appear in this image.

In Fig. 9d, the four methods produce similar segmentation results that do not seem to resemble the cartoon image. Upon closer inspection, the results detect regions of similar pixel intensities that might be difficult to discern from the cartoon image.

In Fig. 10a, the results obtained by Otsu’s method, *k*-means, and local multiphase resemble the cartoon image more than does the multiphase result. The multiphase result attempts to segment the “island” region, but the three methods are able to capture the “island” region on the top right. However, both Otsu’s method and *k*-means are sensitive to intensity inhomogeneities that they capture the border around the “island” region and another region surrounding it. On the other hand, local multiphase does not capture these regions. Additionally, the local multiphase result detects the small streak at the very bottom of the image.

In Fig. 10b, unlike the results from Otsu’s method, *k*-means, and multiphase, the local multiphase result is able to identify the oval region at the right of the image. The oval region has weak edges because of the apparent intensity inhomogeneity, which the other three methods are unable to detect.

As for Fig. 10c, results from the four methods resemble the cartoon image, but local multiphase is able to identify the small “islands” in the middle of the image. The results by Otsu’s method, *k*-means, and multiphase tend to identify larger regions, but for one or two regions of the results, some

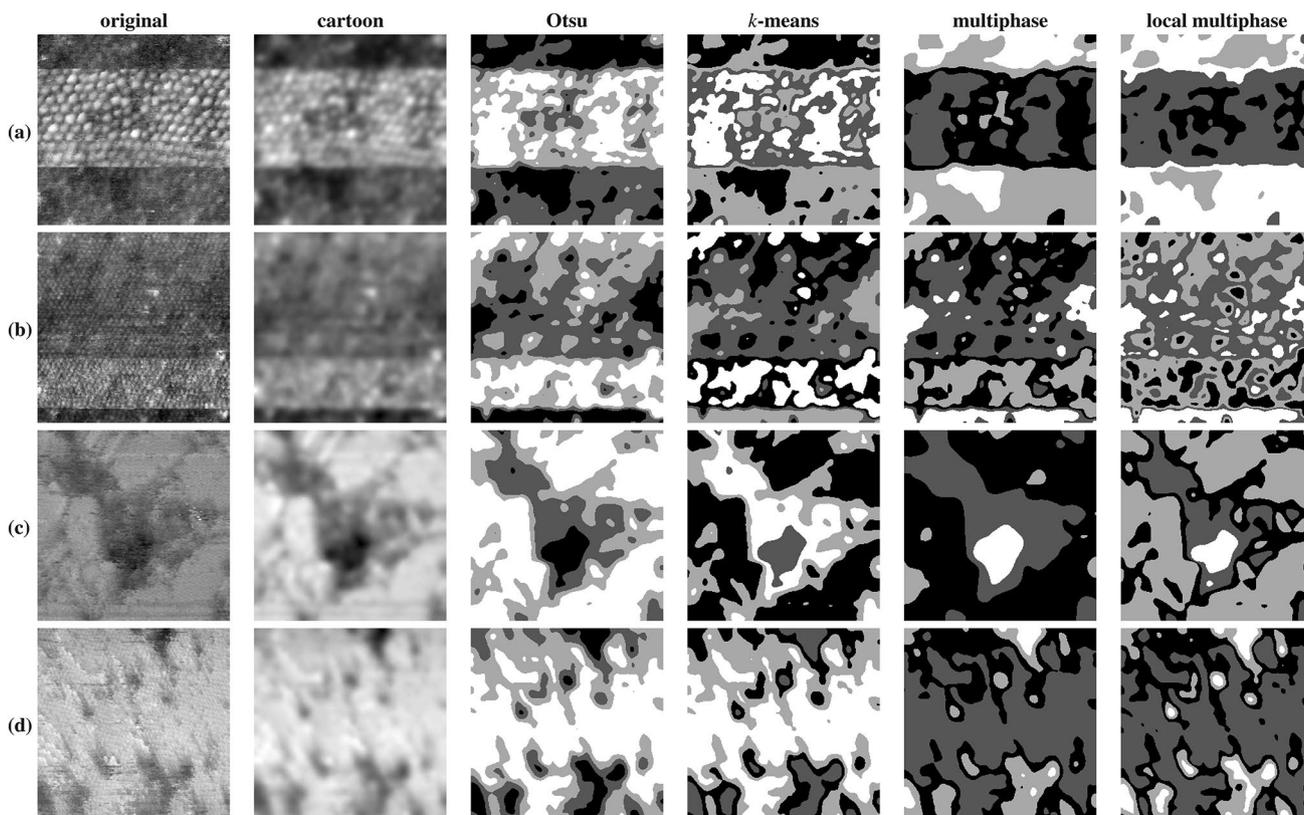


Fig. 11 Comparison between four segmentation methods. Otsu’s method and *k*-means are nonparametric. The parameters for multiphase and local multiphase are: **a** $\lambda = 7$, $\mu = 10^{-3} \times 256^2$, $\beta = 50$, $dt_{CV} = 8$, $dt_{CVloc} = 10$; **b** $\lambda = 5$, $\mu = 10^{-4} \times 256^2$, $\beta = 15$, $dt_{CV} = 0.75$, $dt_{CVloc} = 0.75$; **c** $\lambda = 10$, $\mu = 10^{-3} \times 256^2$, $\beta = 50$,

$dt_{CV} = 20$, $dt_{CVloc} = 2.5$; **d** $\lambda = 10$, $\mu = 10^{-3} \times 256^2$, $\beta = 40$, $dt_{CV} = 0.5$, $dt_{CVloc} = 0.5$. Raw scanning tunneling microscope images of 9,12-carboranedithiol on Au{111}/mica and holey graphene on Au {111}/mica; see [28, 62] for details

parts of one region do not have similar gray-level intensities according to the cartoon image.

In Fig. 10d, four results provide segmentations based on the gray-level intensities. Otsu’s method, *k*-means, and local multiphase are able to detect shadows, such as those in the top left corner and at the bottom of the image, with better precision. The multiphase segmentation captures wider transition regions. However, the slow varying vertical edges are not localized by none of the four methods, in which case, most likely, a non-local version of the segmentation algorithms above would help improve the results.

Figure 11a shows two main textured regions: one “pebbled” and one non-“pebbled.” In the non-“pebbled” regions, the four methods produced similar segmentations. However, the segmentations are different in the pebbled region. The local multiphase segmentation of the pebbled region resembles the pebbled region of the cartoon image more than the results of the other methods. Additionally, it identifies only two regions instead of three regions that are captured by the other three methods.

Figure 11b shows two regions, one rectangular region at the bottom and one surrounding it, thus making up the rest of the image. In the rectangular region, the local multiphase method is able to segment many different regions unidentified by the other methods. Outside the rectangular region, the segmentations between the four methods are similar.

In Fig. 11c, results for Otsu’s method, *k*-means, and local multiphase resemble the cartoon image, whereas the result by multiphase method does not. Hence, multiphase is not robust against the intensity inhomogeneity that is present in the image. Comparing the local multiphase result with the results of the other two methods, we see that the local multiphase result resembles the cartoon image the most because it is able to segment the depressed region, displayed as dark, and another region surrounding it more precisely. In particular, local multiphase is able to segment the surrounding region’s holes, which Otsu’s method and *k*-means fail to do.

In Fig. 11d, the four methods produce results similar to the cartoon image. However, upon closer inspection, the local multiphase result is able to segment certain regions better. For example, it is able to detect the gaps between the

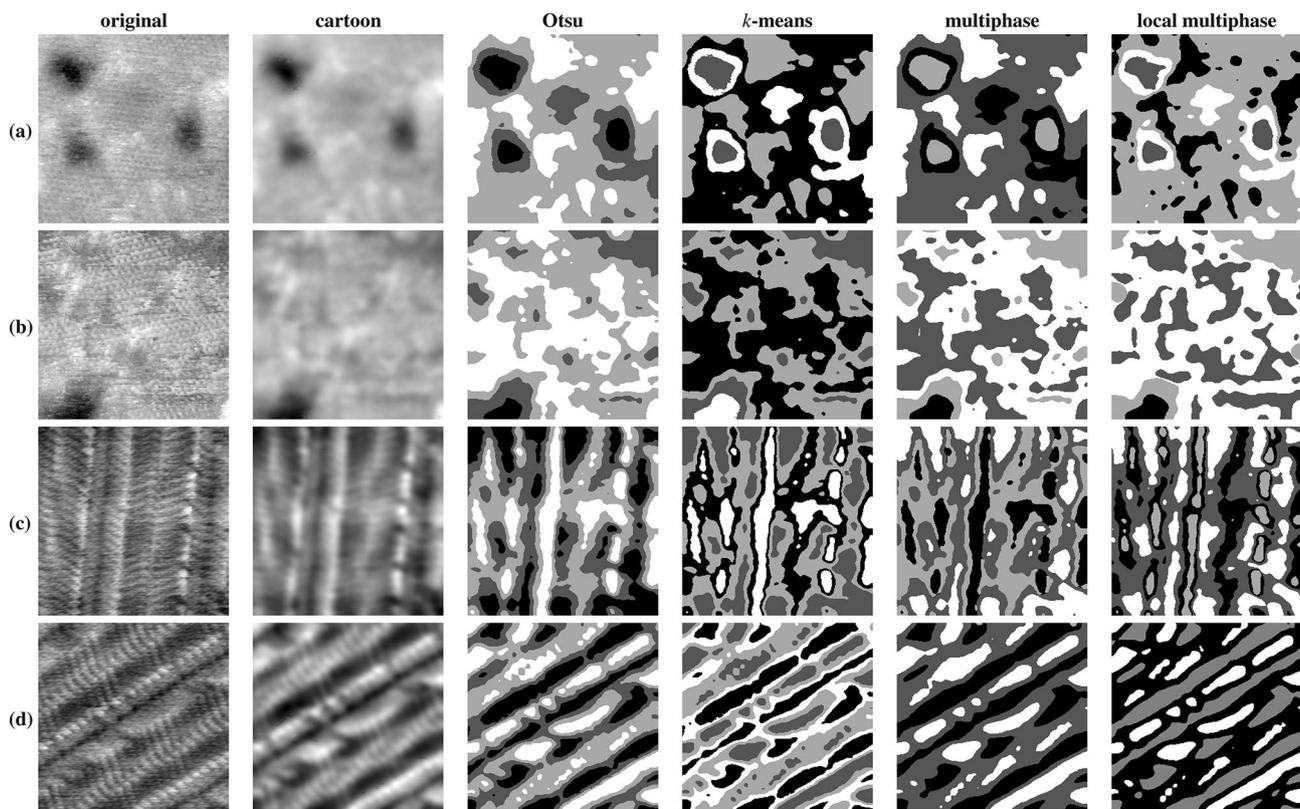


Fig. 12 Comparison between four segmentation methods. Otsu's method and *k*-means are nonparametric. The parameters for multiphase and local multiphase are: **a** $\lambda = 5$, $\mu = 10^{-3} \times 256^2$, $\beta = 50$, $dt_{CV} = 1.0$, $dt_{CVloc} = 1.0$; **b** $\lambda = 7$, $\mu = 10^{-3} \times 256^2$, $\beta = 40$, $dt_{CV} = 0.5$, $dt_{CVloc} = 2.0$; **c** $\lambda = 7$, $\mu = 10^{-4} \times 256^2$, $\beta = 60$, $dt_{CV} = 5$,

$dt_{CVloc} = 15$; **d** $\lambda = 5$, $\mu = 10^{-3} \times 256^2$, $\beta = 15$, $dt_{CV} = 4$, $dt_{CVloc} = 5$. Raw scanning tunneling microscope images of 1ATC9 on Au{111} (see [41, 61] for details) and peptides on highly oriented pyrolytic graphite (reproduced from [69]). Images copyright American Chemical Society

depressed regions at the bottom of the cartoon image and segment them as part of the “background” region. Otsu's method and *k*-means, on the other hand, segment the gaps as the border region surrounding the depressed regions in the cartoon image.

In Fig. 12a, results produced by the four methods are similar. They are able to segment the depressed regions and the surrounding regions. Still, local multiphase seems to segment some of the regions, especially the protruding regions (displayed as bright) in the cartoon image, more precisely than do the other methods.

In Fig. 12b, Otsu's method, *k*-means, and local multiphase produce similar segmentations. The three methods are able to segment the depressed region at the bottom left of the cartoon image, but surrounding it are two consecutive regions. The cartoon image shows that the depressed region is surrounded by only one region, which the local multiphase result does show. In addition, the segmentation of the middle of the image by local multiphase seems most similar to the cartoon image because it is able to detect the slight difference in gray-level intensities.

In Fig. 12c, Otsu's method, *k*-means, and multiphase produce similar segmentations. The three methods are able to capture the white vertical bars present in the cartoon image, but for the bar near the center of the image, the methods fill in the gap it has. Furthermore, they are unable to properly segment the “lightning streak” shown in the cartoon image. On the other hand, the segmentation result by local multiphase is able to segment not only the vertical white bars but also the “lightning streak.” Moreover, the gaps in the white bars are not filled in.

In Fig. 12d, segmentations are similar between Otsu's method and *k*-means, and between multiphase and local multiphase. The four methods are able to segment the bright white, diagonal bars of the cartoon image. However, segmentations of the black, diagonal bars are different between the former two and latter two methods. For the former two methods, some segmentations of the black bars are disconnected when they should not be in the cartoon image. Furthermore, they are surrounded by segmentation of another region, which is not apparently shown in the cartoon image. The results for multiphase and local multiphase do not have these issues. The segmentation by local multiphase is

Table 1 Entropy values of the four segmentation methods on the cartoon image

Image	Metric	Otsu	<i>k</i> -means	Multiphase	Local multiphase
Figure 9a	$H_r(I)$	3.696	3.5940	3.7462	3.9945
	$H_l(I)$	1.0662	1.1595	1.0633	0.7972
	E	4.7622	4.7535	4.8095	4.7917
Figure 9b	$H_r(I)$	3.8440	3.8093	4.4333	4.6816
	$H_l(I)$	1.2679	1.2894	1.0772	1.2551
	E	5.1119	5.0988	5.5105	5.9367
Figure 9c	$H_r(I)$	3.6602	3.6527	3.9928	4.001
	$H_l(I)$	1.1581	1.1537	0.8654	0.8624
	E	4.8183	4.8063	4.8582	4.8625
Figure 9d	$H_r(I)$	3.8647	3.8277	3.8782	3.8618
	$H_l(I)$	1.3318	1.3569	1.3725	1.3734
	E	5.1965	5.1846	5.2507	5.2352
Figure 10a	$H_r(I)$	3.6463	3.6409	3.8547	4.0413
	$H_l(I)$	1.2374	1.2347	1.2216	0.8770
	E	4.8838	4.8756	5.0763	4.9183
Figure 10b	$H_r(I)$	3.6639	3.6502	3.7298	3.9523
	$H_l(I)$	0.9856	0.9945	0.9305	1.2578
	E	4.6494	4.6447	4.6604	5.2101
Figure 10c	$H_r(I)$	3.7417	3.7541	3.7987	4.0233
	$H_l(I)$	1.3649	1.3356	1.3150	1.2466
	E	5.1066	5.0897	5.1137	5.2699
Figure 10d	$H_r(I)$	3.6786	3.6575	3.8461	3.8343
	$H_l(I)$	1.2931	1.3007	1.1423	1.2560
	E	4.9717	4.9583	4.9884	5.0903
Figure 11a	$H_r(I)$	3.8634	3.8587	4.1489	4.2751
	$H_l(I)$	1.3790	1.3708	1.3510	1.2592
	E	5.2424	5.2295	5.4999	5.5343
Figure 11b	$H_r(I)$	3.5508	3.5335	3.5771	3.9523
	$H_l(I)$	1.3138	1.3123	1.2849	1.2578
	E	4.8646	4.8458	4.8620	5.2101
Figure 11c	$H_r(I)$	3.8551	3.8401	4.3150	4.0355
	$H_l(I)$	1.1550	1.1608	0.9234	1.0893
	E	5.0101	5.0008	5.2384	5.1248
Figure 11d	$H_r(I)$	3.5524	3.5038	3.7155	3.8641
	$H_l(I)$	1.2125	1.2447	1.0581	1.1310
	E	4.7649	4.7486	4.7735	4.9951
Figure 12a	$H_r(I)$	3.4642	3.4425	3.4842	3.7980
	$H_l(I)$	1.1294	1.1309	1.1250	1.0464
	E	4.5936	4.5735	4.6092	4.8444
Figure 12b	$H_r(I)$	3.5368	3.4928	3.5336	3.9092
	$H_l(I)$	1.0461	1.0710	1.0608	1.0222
	E	4.5829	4.5638	4.5944	4.9313
Figure 12c	$H_r(I)$	3.7610	3.7458	3.8813	4.2816
	$H_l(I)$	1.3451	1.3451	1.3325	1.2651
	E	5.1061	5.0910	5.2138	5.5467

Table 1 (continued)

Image	Metric	Otsu	<i>k</i> -means	Multiphase	Local multiphase
Figure 12d	$H_r(I)$	3.9653	3.9508	4.3581	4.4424
	$H_l(I)$	1.3569	1.3597	1.0620	0.9897
	E	5.3222	5.3105	5.4200	5.4321

Bold values indicate the best result

slightly better than the segmentation by multiphase because it is more robust against illumination bias present in the image. In particular, the segmentation of the black, diagonal bar at the bottom right is connected in the local multiphase result, while it is not in the multiphase result.

According to Table 1, *k*-means attains the lowest value for the expected region entropy $H_r(I)$ and entropy-based evaluation function E for most of the images, while local multiphase attains the lowest value for layout entropy $H_l(I)$. The reason why *k*-means attains the smallest value of $H_r(I)$ for most images is that it minimizes the sums of the distances between the centroids of the segmentations and their corresponding pixels, or it maximizes uniformity of the pixels of the segmentations. Because the value E is comprised mostly of the value $H_r(I)$, *k*-means therefore attains the smallest value for E for most of the images. Since most of the microscopy images have illumination bias and intensity inhomogeneities, having the smallest $H_r(I)$ and E values does not indicate that the method attains the best segmentation result, and we see that *k*-means produce results dissimilar to their corresponding cartoon images. Local multiphase, on the other hand, attains the largest $H_r(I)$ values for most of the images because the method is robust against illumination bias and intensity inhomogeneities, so segmentations are not uniform in gray-level intensities. However, it does not over-segment for most of the images as indicated by its lowest $H_l(I)$ values.

5.2 Texture segmentation results

In order to segment the texture component of the image, we apply Algorithm 4 to obtain its ECT coefficients. Next, we build the texture feature matrix according to (44). Finally, we apply a clustering algorithm to the energy matrix. The number of clusters is determined by the user. In our experiments, the clustering algorithms we use are *k*-means, already implemented in MATLAB, and multiclass MBO clustering [27, 45]. Both methods use the cityblock metric to measure

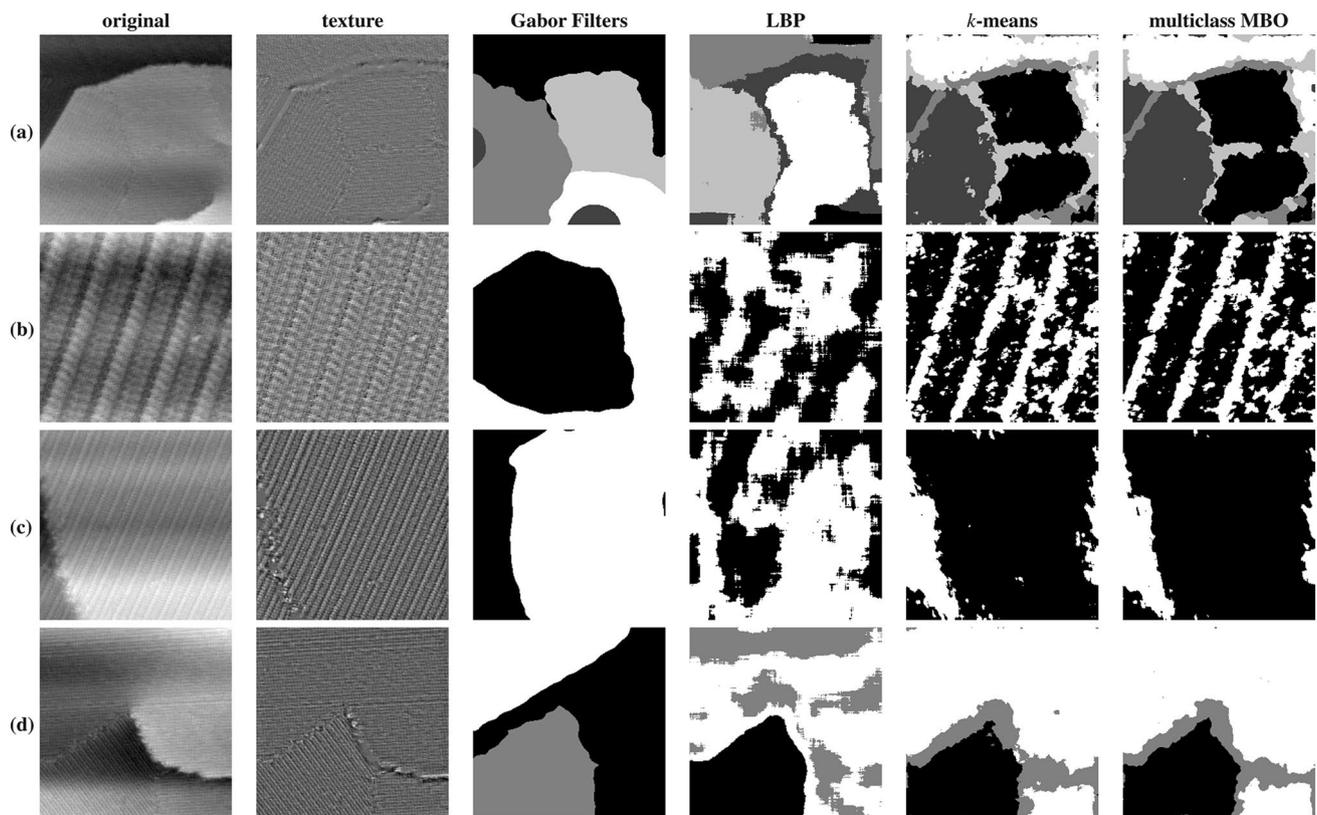


Fig. 13 Texture segmentation results. The parameters for *k*-means and multiclass MBO are: **a** $\tau = 92$ nd percentile, $k = 5$, $dt = 0.03$. **b** $\tau = 99.50$ th percentile, $k = 2$, $dt = 0.10$. **c** $\tau = 98.80$ th percentile, $k = 2$, $dt = 0.05$. **d** $\tau = 85$ th percentile, $k = 3$, $dt = 0.05$. Raw scan-

ning tunneling microscope images of cyanide on Au{111}, reproduced from [35] with permission. Texture segmentation results in (a) were reproduced with permission from [35]. Images and segmentation results copyright American Chemical Society

the similarity between data points of the energy matrix in order to determine the clusters. As an unsupervised method, *k*-means utilizes random initialization to determine the initial centroids. In order to obtain high-quality clusters, Arthur and Vassilvitskii devised a heuristic to ensure that every two initial centroids are dissimilar to each other [4]. As a semi-supervised method, the multiclass MBO clustering randomly selects 25% of the labels determined by *k*-means as its initialization. Note that because of random initialization of the centroids, results may differ for every run. Hence, for each image, we run 10 replications of *k*-means and select the best result based on minimum within-cluster sums of point-to-centroid distances. Only the best result is shown and is used as an initialization to multiclass MBO clustering.

The multiclass MBO is a graph-based method, so it requires constructing the graph Laplacian matrix, which is expensive both in computation and in memory. Instead of computing the matrix exactly, we use Nyström extension to approximate its eigen-decomposition [25]. The method requires from the user the number of data points to sample and the number of eigenvectors to compute. In our

experiments, we sample 300 data points and compute 30 eigenvectors.

For all images, the multiclass MBO clustering is set with the following parameters: $\mu = 30$ (fidelity parameter) and $\eta = 10^{-7}$ (tolerance), following the notations in [27]. The thresholding step is performed after every three iterations of the diffusion step. The time step dt is the only parameter that differs between the images. It is set below 0.20.

We compare our proposed methods to texture segmentation algorithms that use Gabor filters¹ [39] and local binary pattern (LBP)² [53]. The results are shown in Figs. 13, 14, 15, and 16. Note that no ground truth is available for the texture images. Moreover, there does not exist an unsupervised metric for texture segmentation, especially for micro-textures. Although we cannot validate the results quantitatively, we observe that texture segmentations by Gabor filters do not resemble their corresponding texture images at all, while some of the texture segmentations by LBP somewhat

¹ <https://www.mathworks.com/help/images/texture-segmentation-using-gabor-filters.html>

² <https://www.mathworks.com/matlabcentral/fileexchange/52753-kolian1-texture-segmentation-lbp-vs-gldm>

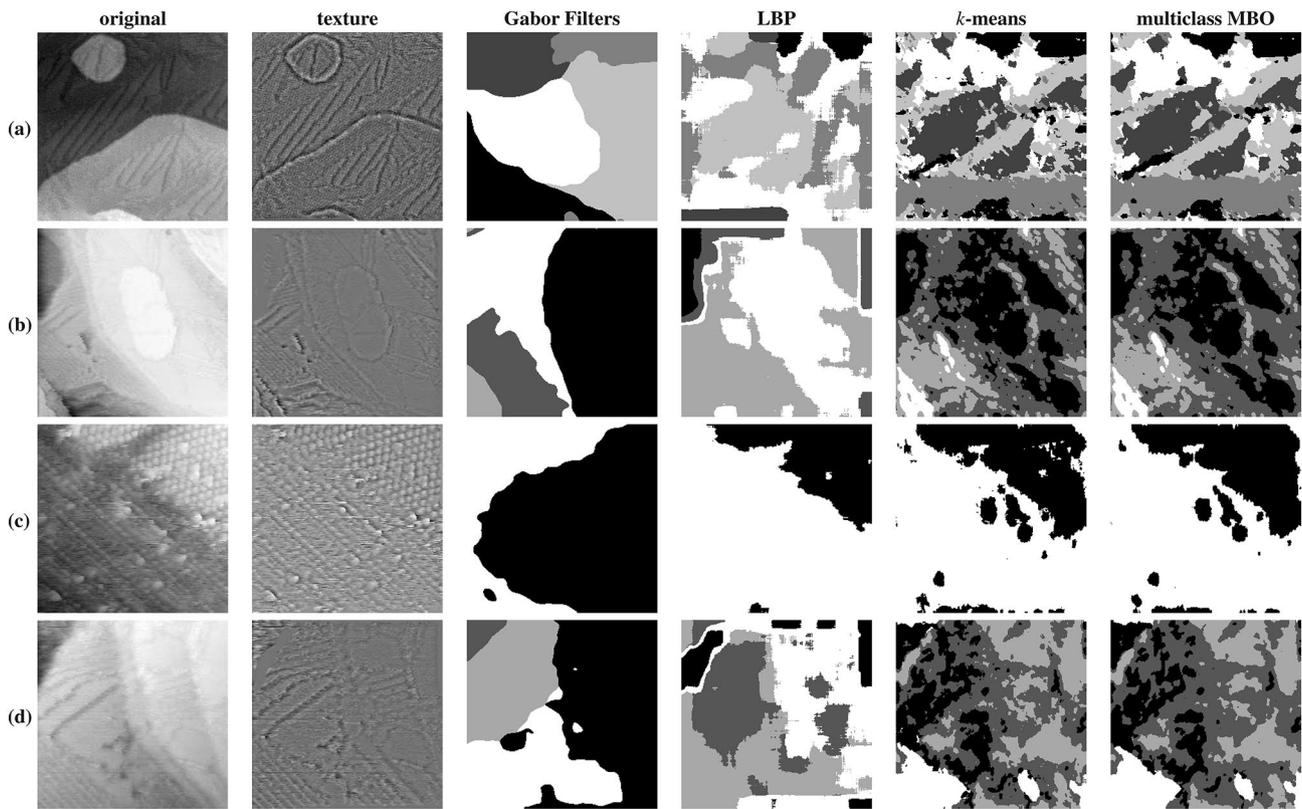


Fig. 14 Texture segmentation results. The parameters for *k*-means and multiclass MBO are **a** $\tau = 95.15$ th percentile, $k = 5$, $dt = 0.10$; **b** $\tau = 45$ th percentile, $k = 4$, $dt = 0.10$; **c** $\tau = 85$ th percentile, $k = 2$, $dt = 0.05$; **d** $\tau = 72.50$ th percentile, $k = 4$, $dt = 0.05$. Raw scanning

tunneling microscope images of cyanide on Au{111}, reproduced from [35] with permission. Images copyright American Chemical Society

resemble their corresponding texture images. On the other hand, most of the segmentation results from our proposed methods are able to identify the recognizable texture patterns of the images. Although the results of the proposed methods are similar, the results obtained from multiclass MBO clustering tend to have smoother clusters and appear cleaner than the results obtained from *k*-means.

In Fig. 13a, we select $k = 5$. For the four methods, the three clusters correspond to the periodic texture patterns: one that appears to have horizontal direction (the middle cluster), one that is slanted upward (the left cluster), and one that is finer and is slanted downward (top cluster). The proposed *k*-means and multiclass MBO identify two of the clusters as the material interfaces: One interface is a deep indentation and the other appears perforated. Gabor filters were not able to cluster them; LBP identify the two interfaces, but not entirely, as one cluster.

In Fig. 13b, we select $k = 2$. The result produced from Gabor filters does not resemble the texture image. For the other three methods, two clusters were identified: One cluster corresponds to the perforated lines, while the other corresponds to the regions between the lines. The result produced by LBP appears blotchy. On the other hand, results

by *k*-means and multiclass MBO resemble the texture image better than the LBP result. However, some of the pixels appear to be misclassified. For example, pixels in the region between the perforated lines are classified as the same cluster. This result may be attributed to the noise or unexpected anomalies within the texture patterns. Since the clustering result from multiclass MBO appears to have less misclassified pixels than does *k*-means, it affirms that multiclass MBO is more resistant to noise, even with 25% of the labels provided by *k*-means.

In Fig. 13c, we select $k = 2$ again since there two texture patterns: One corresponds to the uniform texture pattern that has an upward, slanted direction, while the other corresponds to the break dividing the uniform texture pattern. Both the Gabor filters and LBP results fail to cluster the two texture patterns. However, the proposed methods are able to cluster the two texture patterns, but we observe that pixels near the break tend to be misclassified because the break is relatively small compared to the uniform texture pattern.

In Fig. 13d, we select $k = 3$: one corresponding to the perforated or indented edge, another one corresponding to the slightly horizontal texture pattern (top and bottom right), and the last one corresponding to the texture pattern slanted

downward (bottom left). The method that uses Gabor filters is able to identify the bottom left texture, while LBP is able to identify the slanted and horizontal texture patterns. Both proposed methods identify the three patterns seen in the texture image. Again, like for the previous image discussed, pixels near the edge tend to be misclassified.

In Fig. 14a, the methods have difficulty clustering the texture patterns because the texture image itself has various texture patterns of different sizes in various locations. The clustering results by Gabor filters and LBP do not seem to resemble the texture image at all. The clustering results by the proposed methods seem to identify some of the texture patterns. For example, two of the clusters correspond to texture patterns that slant upward, but one has finer texture, while the other has sharper edges. The third cluster identifies with the horizontal texture pattern (top and bottom cluster).

In Fig. 14b, we select $k = 4$. Both clustering results by Gabor filters and LBP do not seem to provide meaningful identification of the texture patterns in the texture image. In the results by the proposed methods, one cluster corresponds to the smoothest regions of the texture component. Another corresponds to the texture pattern slanted downward and with rough edges (bottom left cluster). The third cluster corresponds to the finer texture pattern slanted downward (top right, bottom left, and along the right edge of the elliptical smooth region). However, the last cluster appears to correspond to the miscellaneous region that does not have any apparent pattern.

In Fig. 14c, we select $k = 2$: one corresponding to the scaly texture pattern in the top right and the other one corresponding to the rough texture pattern that have some lines slanted downward. The clustering result by Gabor filters fails to cluster the two texture patterns properly, whereas the other three methods are able to do so. For the proposed methods, some pixels outside of the top right region belong to the same cluster, and most of them resemble the top right texture pattern as they look like scales.

In Fig. 14d, we select $k = 4$. Both clustering results by Gabor filters and LBP do not appear to identify any texture patterns found in the texture image. The proposed methods, on the other hand, are able to do so for some of the clusters. One cluster corresponds to the texture pattern slanted downward, located at the bottom left and bottom middle of the image. Another cluster corresponds to the smooth regions, located at the top left corner and most of the right side of the image. The other cluster seems to correspond to the rough edges, located mostly in the left of the image.

In Fig. 15a-b, we select $k = 2$ because there are two apparent texture patterns. Clustering by Gabor filters fails to identify the two patterns in both images. Clustering by LBP captures the two texture patterns in both images, but in Fig. 15b, the result has considerable amount of misclassification. For Fig. 15a, the results of our proposed method are comparable to the LBP result. However, for Fig. 15b, they have much less obvious misclassification.

In Fig. 15c, we select $k = 3$. According to the Gabor filters result, the three texture patterns that were clustered are the scaly texture pattern on the top right, the ragged texture pattern in the center of the image, and the miscellaneous texture pattern at the bottom. However, the texture pattern in the bottom right of the texture image appears similar to the texture pattern in the top right, which should have been clustered together. The result by LBP is able to cluster the bottom right texture pattern but not together entirely with the top right texture pattern. It also clusters the ragged texture pattern in the center of the image. The third cluster corresponds to the various other texture patterns. The results of the proposed methods are able to group together the scaly texture pattern in the right side of the image. Moreover, they show a more precise shape of the ragged texture pattern in the center of the image. The third cluster seems to be a miscellaneous texture pattern.

In Fig. 15d, we select $k = 3$. None of the clusters in the Gabor filters result appears to correspond to any particular texture pattern in the image. The clusters of the LBP result appear to identify the scaly texture pattern on the right side of the image and the ragged texture pattern at the bottom. The results by the proposed methods identify two of the same texture patterns, but the cluster shapes appear to be more precise than the cluster shapes of the LBP result. The last cluster seems to correspond to a miscellaneous texture pattern.

In Fig. 16a, we select $k = 3$. The Gabor filters result does not identify any particular texture pattern. The other three methods are able to cluster two apparent texture patterns in the image: the scaly texture pattern (e.g., top right and center of texture image) and the ragged texture pattern (e.g., right and middle left of the texture image). The clustering results by the proposed methods seem to have more precise clusters than do the results of LBP.

In Fig. 16b, we select $k = 3$. The Gabor filters result does not identify any particular texture pattern. The proposed methods are able to identify two texture patterns: the scaly texture pattern and one at the bottom left of the texture image. The result by LBP identifies the same two texture patterns, but the cluster shapes seem to be more precise.

In Fig. 16c-d, we select $k = 4$. For both figures, Gabor filters fail to identify any meaningful texture patterns. The results by LBP are able to identify some texture patterns. Figure 16c shows four slightly diagonal lines in the center of the texture image. LBP is able to cluster the three left lines, but not in its entirety, as two separate groups. The fourth line is not clustered. The texture patterns between the lines are not clustered properly. For Fig. 16d, LBP is not able to cluster the diagonal, “perforated” lines. However, the clustering results of the proposed method resemble their corresponding texture images more than the clustering results of the former two methods. For Fig. 16c, the proposed methods identify

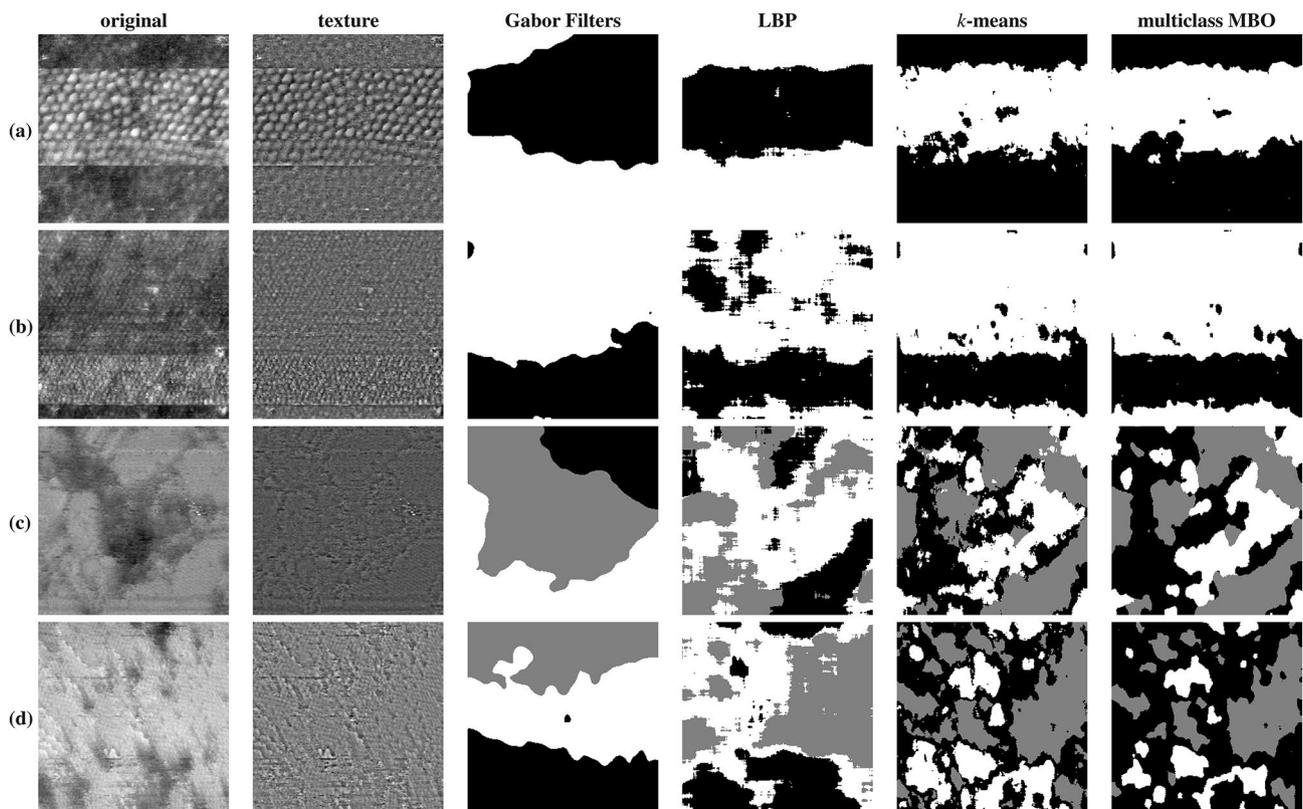


Fig. 15 Texture segmentation results. The parameters for k -means and multiclass MBO are **a** $\tau = 75$ th percentile, $k = 2$, $dt = 0.10$; **b** $\tau = 90$ th percentile, $k = 2$, $dt = 0.05$; **c** $\tau = 40$ th percentile, $k = 3$,

$dt = 0.025$; **d** $\tau = 87.50$ th percentile, $k = 3$, $dt = 0.01$. Raw scanning tunneling microscope images of 9,12-carboranedithiol on Au{111}/mica and holey graphene on Au {111}/mica, see [28, 62] for details

the four lines and the slanted texture patterns between them. For Fig. 16d, they identify two perforated diagonal lines as one cluster. In addition, their results reveal alternating texture patterns in the texture image.

6 Conclusions

In this paper, we proposed a framework to segment STM images, combining variational methods and a clustering algorithm based on features extracted by the empirical wavelet transform. The expected information of microscopy images led us to first apply a cartoon + texture decomposition and then run a modified version of the multiphase CV model on the cartoon part, and a clustering of features extracted by the empirical curvelet transform on the texture part. The results in Sect. 5 demonstrated the efficacy of this framework to analyze STM images. In [35], Guttentag et al. already used the proposed approach to characterize patterns of cyanide molecules on Au{111}, complementing the results in another related work [36].

There are several directions to investigate in order to improve the proposed framework. One direction is employing

directional filters [9, 10] instead of isotropic filters L_σ in the cartoon + texture decomposition, which are designed to separate noise and micro-texture better in the proximity of edges. As such, in the proposed local MCV model, the Gaussian filter could be upgraded to a non-local filter, incorporating direction to better characterize weak edges. To gain robustness against low-contrast and impulse noise, one could work with the L^1 fidelity term, but since the latter is non-differentiable at the origin, one would have to use convex optimization algorithms like primal–dual methods. The application of the entropy-based evaluation criteria to our results shows that the metric is sensitive to illumination bias and intensity inhomogeneity, which means that the expected region entropy may need to be modified to account for them, layout entropy needs to be weighed more, or both. Finally, in order to overcome artifacts (e.g., scars, occlusions, scratches), one could incorporate an indicator function as done in [72]. As for the segmentation of the texture component, the use of other features than the local energy of the curvelet coefficient, such as the co-occurrence matrices [37] constructed from the empirical curvelets, is likely to lead to more adapted clustering results. Furthermore, we can incorporate more empirical wavelet features reviewed in [38].

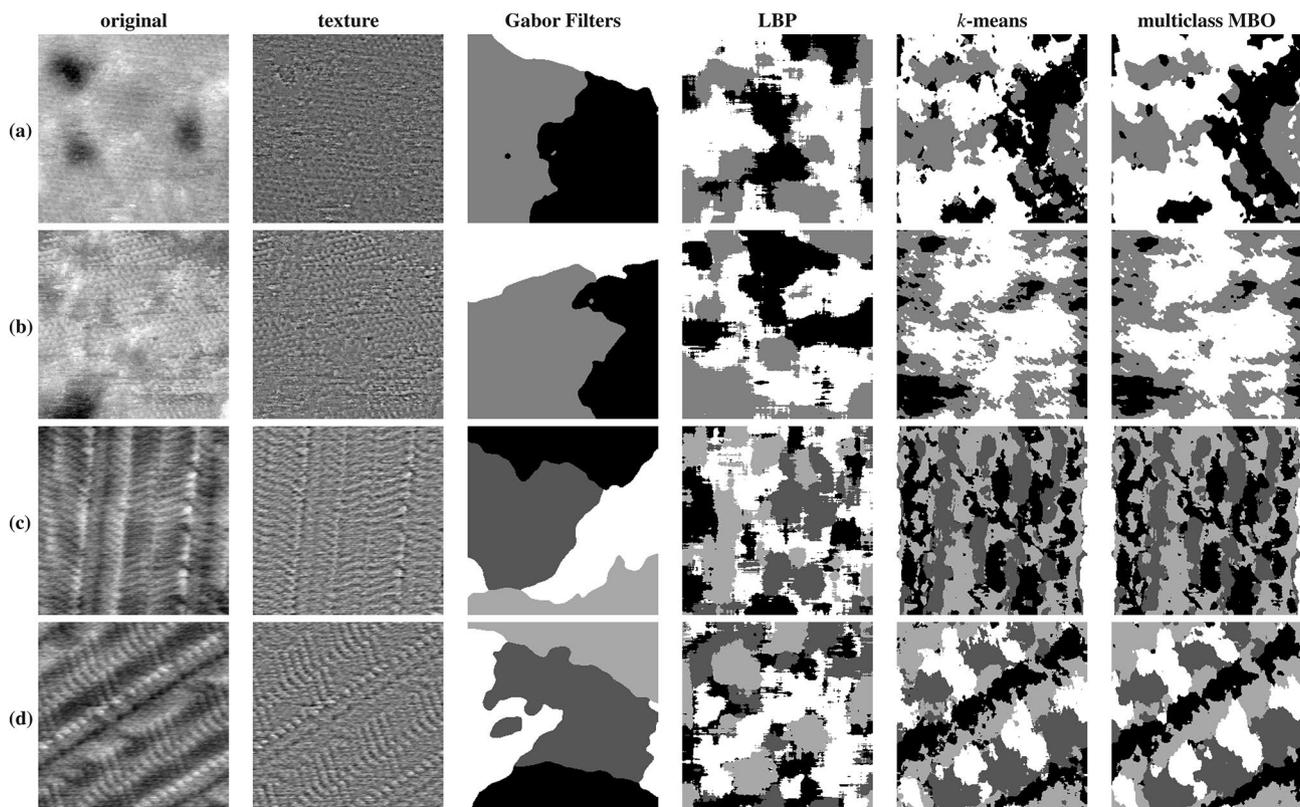


Fig. 16 Texture segmentation results. The parameters for k -means and multiclass MBO are **a** $\tau = 75$ th percentile, $k = 3$, $dt = 0.05$; **b** $\tau = 35$ th percentile, $k = 3$, $\tau = 0.025$; **c** $\tau = 50$ th percentile, $k = 4$, $dt = 0.15$; **d** $\tau = 76.50$ th percentile, $k = 4$, $dt = 0.05$. Raw scanning

tunneling microscope images of 1ATC9 on Au{111} (see [41, 61] for details) and peptides on highly oriented pyrolytic graphite (reproduced from [69]). Images copyright American Chemical Society

Overall, the proposed framework has produced remarkable segmentation results of STM images using variants of the state-of-the-art image processing algorithms. One could modify and apply this framework to other kinds of images, thus leading to more interesting contributions and applications in other scientific areas beyond nanoscience.

Acknowledgements This work was supported by the W.M. Keck Foundation Center for Leveraging Sparsity. The self-assembly and STM imaging were supported by the US Department of Energy (Grant #DE-SC-1037004). L. Torres Mandiola was partially supported by NSF DMS-1312361. K. Bui, J. Fauman, and D. Kes were partially supported by NSF DMS-1045536. A. Bertozzi was supported by Simons Math + X Investigator Award Number 510776. The authors thank the editor and anonymous reviewers for their helpful comments in improving the quality of the manuscript.

References

- Ambrosio L, Tortorelli VM (1990) Approximation of functional depending on jumps by elliptic functional via t -convergence. *Commun Pure Appl Math* 43(8):999–1036
- Arivazhagan S, Ganesan L (2003) Texture segmentation using wavelet transform. *Pattern Recognit Lett* 24(16):3197–3203
- Arivazhagan S, Ganesan L, Kumar TS (2006) Texture classification using curvelet statistical and co-occurrence features. In: *ICPR 2006. 18th international conference on pattern recognition, 2006, vol 2*, pp 938–941. IEEE
- Arthur D, Vassilvitskii S (2007) K-means++: the advantages of careful seeding. In: *Proceedings of the 18th annual ACM-SIAM symposium on discrete algorithms*
- Averbuch A, Coifman RR, Donoho DL, Elad M, Israeli M (2006) Fast and accurate polar Fourier transform. *Appl Comput Harmon Anal* 21(2):145–167
- Averbuch A, Coifman RR, Donoho DL, Israeli M, Shkolnitsky Y (2008) A framework for discrete integral transformations I—the pseudopolar Fourier transform. *SIAM J Sci Comput* 30(2):764–784
- Boyd S, Parikh N, Chu E, Peleato B, Eckstein J et al (2011) Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found Trends® Mach Learn* 3(1):1–122
- Buades A, Le T, Morel JM, Vese L (2011) Cartoon + texture image decomposition. *Image processing on line 1*
- Buades A, Lisani JL (2016) Directional filters for cartoon + texture image decomposition. *Image Process Line* 6:75–88
- Buades A, Lisani JL (2016) Directional filters for color cartoon + texture image and video decomposition. *J Math Imag Vis* 55(1):125–135

11. Bumm LA, Arnold JJ, Cygan MT, Dunbar TD, Burgin TP, Jones L, Allara DL, Tour JM, Weiss PS (1996) Are single molecular wires conducting? *Science* 271:1705–1707
12. Candes E, Demanet L, Donoho D, Ying L (2006) Fast discrete curvelet transforms. *Multiscale Model Simul* 5(3):861–899
13. Candes EJ, Donoho DL (2005) Continuous curvelet transform: I. Resolution of the wavefront set. *Appl Comput Harmon Anal* 19(2):162–197
14. Chambolle A (2004) An algorithm for total variation minimization and applications. *J Math Imag Vis* 20(1):89–97
15. Chan TF, Esedoglu S (2005) Aspects of total variation regularized L_1 function approximation. *SIAM J Appl Math* 65:1817–1837
16. Chan TF, Sandberg BY, Vese LA (2000) Active contours without edges for vector-valued images. *J Vis Commun Image Represent* 11(2):130–141
17. Claridge SA, Liao WS, Thomas JC, Zhao Y, Cao HH, Cheunkar S, Serino AC, Andrews AM, Weiss PS (2013) From the bottom up: dimensional control and characterization in molecular monolayers. *Chem Soc Rev* 42(7):2725–2745
18. Dameron AA, Charles LF, Weiss PS (2005) Structures and displacement of 1-Adamantanethiol self-assembled monolayers on Au 111. *J Am Chem Soc* 127(24):8697–8704
19. Dragomireskiy K, Zosso D (2014) Variational mode decomposition. *IEEE Trans Signal Process* 62(3):531–544
20. Dragomireskiy K, Zosso D (2015) Two-dimensional variational mode decomposition. In: *International workshop on energy minimization methods in computer vision and pattern recognition*, pp 197–208. Springer
21. Dunn D, Higgins WE (1995) Optimal gabor filters for texture segmentation. *IEEE Trans Image Process* 4(7):947–964
22. Dunn D, Higgins WE, Wakeley J (1994) Texture segmentation using 2-d gabor elementary functions. *IEEE Trans Pattern Anal Mach Intell* 16(2):130–149
23. Dunn D.F, Higgins W.E (1993) Optimal gabor-filter design for texture segmentation. In: *1993 IEEE international conference on acoustics, speech, and signal processing, 1993. ICASSP-93.*, vol 5, pp 37–40. IEEE
24. Esedoglu S, Tsai YHR (2006) Threshold dynamics for the piecewise constant Mumford-Shah functional. *J Comput Phys* 211(1):367–384
25. Fowlkes C, Belongie S, Chung F, Malik J (2004) Spectral grouping using the Nyström method. *IEEE Trans Pattern Anal Mach Intell* 26(2):214–225
26. Gabay D, Mercier B (1976) A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Comput Math Appl* 2(1):17–40
27. Garcia-Cardona C, Merkurjev E, Bertozzi AL, Flenner A, Percus AG (2014) Multiclass data segmentation using diffuse interface methods on graphs. *IEEE Trans Pattern Anal Mach Intell* 36(8):1600–1613
28. Gethers ML, Thomas JC, Jiang S, Weiss NO, Duan X, Goddard WA III, Weiss PS (2015) Holey graphene as a weed barrier for molecules. *ACS Nano* 9(11):10909–10915
29. Getreuer P (2012) Chan-Vese segmentation. *Image Process Line* 2:214–224
30. Gilles J (2013) Empirical wavelet transform. *IEEE Trans Signal Process* 61(16):3999–4010. <https://doi.org/10.1109/TSP.2013.2265222>
31. Gilles J, Heal K (2014) A parameterless scale-space approach to find meaningful modes in histograms-application to image and spectrum segmentation. *Int J Wavel Multiresolut Inf Process* 12(6):14500441–14500447. <https://doi.org/10.1142/S0219691314500441>
32. Gilles J, Tran G, Osher S (2014) 2D empirical transforms. Wavelets, ridgelets and curvelets revisited. *SIAM J Imag Sci* 7(1):157–186. <https://doi.org/10.1137/130923774>
33. Glowinski R, Marroco A (1975) Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéaires. *Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique* 9(R2): 41–76
34. Gooding JJ, Mearns F, Yang W, Liu J (2003) Self-assembled monolayers into the 21st century: recent advances and applications. *Electroanalysis* 15(2):81–96
35. Guttentag AI, Barr KK, Song TB, Bui KV, Fauman JN, Torres LF, Kes DD, Ciomaga A, Gilles J, Sullivan NF, Yang Y, Allara DL, Zharnikov M, Weiss PS (2016) Hexagons to ribbons: flipping cyanide on Au {111}. *J Am Chem Soc* 138(48):15580–15586
36. Guttentag AI, Wachter T, Barr KK, Abendroth JM, Song TB, Sullivan NF, Yang Y, Allara DL, Zharnikov M, Weiss PS (2016) Surface structure and electron transfer dynamics of the self-assembly of cyanide on Au {111}. *J Phys Chem C* 120(47):26736–26746
37. Haralick RM, Shanmugam K, Dinstein I (1973) Textural features for image classification. *IEEE Trans Syst Man Cybern* 3(6):610–621
38. Huang Y, De Bortoli V, Zhou F, Gilles J (2018) Review of wavelet-based unsupervised texture segmentation, advantage of adaptive wavelets. *IET Image Process* 12(9):1626–1638
39. Jain AK, Farrokhnia F (1991) Unsupervised texture segmentation using gabor filters. *Pattern Recognit* 24(12):1167–1186
40. Kim M, Hohman JN, Cao Y, Houk KN, Ma H, Jen AKY, Weiss PS (2011) Creating favorable geometries for directing organic photoreactions in alkanethiolate monolayers. *Science* 331(6022):1312–1315
41. Kim M, Hohman JN, Serino AC, Weiss PS (2010) Structural manipulation of hydrogen-bonding networks in amide-containing alkanethiolate monolayers via electrochemical processing. *J Phys Chem C* 114(46):19744–19751
42. Labate D, Lim W-Q, Kutyniok G, Weiss G (2005) Sparse multi-dimensional representation using shearlets. In: *Wavelets XI*, vol 5914. International Society for Optics and Photonics
43. Love JC, Estroff LA, Kriebel JK, Nuzzo RG, Whitesides GM (2005) Self-assembled monolayers of thiols on metals as a form of nanotechnology. *Chem Rev* 105(4):1103–1170
44. March R (1992) Visual reconstruction with discontinuities using variational methods. *Image Vis Comput* 10(1):30–38
45. Merkurjev E, Garcia-Cardona C, Bertozzi AL, Flenner A, Percus AG (2014) Diffuse interface methods for multiclass segmentation of high-dimensional data. *Appl Math Lett* 33:29–34. <https://doi.org/10.1016/j.aml.2014.02.008>
46. Merriman B, Bence J.K, Osher S (1992) Diffusion generated motion by mean curvature. In: *Proceedings of the geometry center workshop*
47. Merriman B, Bence JK, Osher SJ (1994) Motion of multiple junctions: a level set approach. *J Comput Phys* 112(2):334–363
48. Meyer Y (2001) *Oscillating Patterns in Image Processing and Nonlinear Evolution Equations: The Fifteenth Dean Jacqueline B. Lewis Memorial Lectures*. American Mathematical Society, Boston, MA, US
49. Modica L (1987) The gradient theory of phase transitions and the minimal interface criterion. *Arch Ration Mech Anal* 98(2):123–142
50. Modica L, Mortola S (1977) Un esempio di Γ -convergenza. *Bollettino della Unione Matematica Italiana B* 14(5):285–299
51. Mumford D, Shah J (1989) Optimal approximations by piecewise smooth functions and associated variational problems. *Commun Pure Appl Math* 42(5):577–685
52. Nuzzo RG, Allara DL (1983) Adsorption of bifunctional organic disulfides on gold surfaces. *J Am Chem Soc* 105(13):4481–4483
53. Ojala T, Pietikainen M, Maenpaa T (2002) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans Pattern Anal Mach Intell* 24(7):971–987

54. Osher S, Sethian JA (1988) Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *J Comput Phys* 79(1):12–49
55. Otsu N (1979) A threshold selection method from gray-level histograms. *IEEE Trans Syst Man Cybern* 9(1):62–66
56. Poirier GE (1997) Characterization of organosulfur molecular monolayers on Au (111) using scanning tunneling microscopy. *Chem Rev* 97(4):1117–1128
57. Rudin LI, Osher S, Fatemi E (1992) Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenom* 60(1–4):259–268
58. Shen L, Yin Q (2009) Texture classification using curvelet transform. In: *Proceedings of the international symposium on information processing (ISIP09)*, pp 319–324. Citeseer
59. Smith RK, Lewis PA, Weiss PS (2004) Patterning self-assembled monolayers. *Progress Surf Sci* 75(1):1–68
60. Strang G (1993) Wavelet transforms versus fourier transforms. *Bull Am Math Soc* 28(2):288–305
61. Thomas JC, Goronzy DP, Dragomiretskiy K, Zosso D, Gilles J, Osher SJ, Bertozzi AL, Weiss PS (2016) Mapping buried hydrogen-bonding networks. *ACS Nano* 10(5):5446–5451
62. Thomas JC, Goronzy DP, Serino AC, Auluck HS, Irving OR, Jimenez-Izal E, Deirmenjian JM, Machacek J, Sautet P, Alexandrova AN et al (2018) Acid-base control of valency within carbonatedithiol self-assembled monolayers: molecules do the can-can. *ACS Nano* 12(3):2211–2221
63. Thomas JC, Schwartz JJ, Hohman JN, Claridge SA, Auluck HS, Serino AC, Spokoyny AM, Tran G, Kelly KF, Mirkin CA, Gilles J, Osher SJ, Weiss PS (2015) Defect-tolerant aligned dipoles within two-dimensional plastic lattices. *ACS Nano* 9(5):4734–4742
64. Unser M (1995) Texture classification and segmentation using wavelet frames. *IEEE Trans Image Process* 4(11):1549–1560
65. Vese LA, Chan TF (1997) Reduced non-convex functional approximations for image restoration and segmentation. *UCLA CAM Reports* pp 97–56
66. Vese LA, Chan TF (2002) A multiphase level set framework for image segmentation using the Mumford and Shah model. *Int J Comput Vis* 50(3):271–293
67. Wang XF, Huang DS, Xu H (2010) An efficient local Chan-Vese model for image segmentation. *Pattern Recognit* 43(3):603–618
68. Weldon TP, Higgins WE, Dunn DF (1996) Efficient gabor filter design for texture segmentation. *Pattern Recognit* 29(12):2005–2015
69. Yugay D, Goronzy DP, Kawakami LM, Claridge SA, Song TB, Yan Z, Xie YH, Gilles J, Yang Y, Weiss PS (2016) Copper ion binding site in β -amyloid peptide. *Nano Lett* 16(10):6282–6289
70. Zhang H, Fritts JE, Goldman S.A (2003) An entropy-based objective evaluation method for image segmentation. In: *Storage and Retrieval Methods and Applications for Multimedia 2004*, vol 5307, pp. 38–50. International Society for Optics and Photonics
71. Zhang H, Fritts JE, Goldman SA (2008) Image segmentation evaluation: a survey of unsupervised methods. *Comput Vis Image Underst* 110(2):260–280
72. Zosso D, An J, Stevick J, Takaki N, Weiss M, Slaughter LS, Cao HH, Weiss PS, Bertozzi AL (2017) Image segmentation with dynamic artifacts detection and bias correction. *Inverse Probl Imaging* 11(3):577–600
73. Zosso D, Dragomiretskiy K, Bertozzi AL, Weiss PS (2017) Two-dimensional compact variational mode decomposition. *J Math Image Vis* 58(2):294–320

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Kevin Bui¹  · Jacob Fauman² · David Kes³ · Leticia Torres Mandiola⁴ · Adina Ciomaga⁵ · Ricardo Salazar⁶ · Andrea L. Bertozzi⁶ · Jérôme Gilles⁷ · Dominic P. Goronzy^{8,9}  · Andrew I. Guttentag^{8,9} · Paul S. Weiss^{8,9,10}

Jacob Fauman
jacob.fauman@colorado.edu

David Kes
ddkes@dons.usfca.edu

Leticia Torres Mandiola
torres.leticia9@gmail.com

Adina Ciomaga
adina@ljl.univ-paris-diderot.fr

Ricardo Salazar
rsalazar@math.ucla.edu

Andrea L. Bertozzi
bertozzi@math.ucla.edu

Jérôme Gilles
jgilles@sdsu.edu

Dominic P. Goronzy
goronzy@chem.ucla.edu

Andrew I. Guttentag
aguttentag@ucla.edu

Paul S. Weiss
psw@cnsi.ucla.edu

¹ Department of Mathematics, University of California, Irvine, 340 Rowland Hall, Irvine, CA 92697-3875, USA

² Department of Applied Mathematics, 526 UCB, ECOT 225, University of Colorado Boulder, Boulder, Colorado 80309-0526, USA

³ College of Arts and Sciences, University of San Francisco, USF Downtown Campus, 101 Howard Street, San Francisco, CA 94105, USA

⁴ Department of Mathematics, South Kensington Campus, Imperial College London, London SW7 2AZ, UK

⁵ Sorbonne Paris Cité, Laboratoire Jacques-Louis Lions, UMR 7598, UPMC, CNRS, University Paris Diderot, 75205 Paris, France

⁶ Department of Mathematics, University of California, Los Angeles, 520 Portola Plaza, Los Angeles, CA 90095-1555, USA

⁷ Department of Mathematics and Statistics, San Diego State University, 5500 Campanile Dr, San Diego, CA 92181-7720, USA

⁸ Department of Chemistry and Biochemistry, University of California, Los Angeles, Los Angeles, California 90095, USA

⁹ California NanoSystems Institute, University of California, Los Angeles, Los Angeles, California 90095, USA

¹⁰ Department of Materials Science and Engineering, University of California, Los Angeles, Los Angeles, California 90095, USA