J. Micromech. Microeng. **17** (2007) S27–S37 doi:10.1088/0960-1317/17/4/S02

Faster simulations of step bunching during anisotropic etching: formation of zigzag structures on Si(1 1 0)

M A Gosálvez[1], Y Xing[1], T Hynninen[2], M Uwaha[3], A S Foster[2], R M Nieminen[2] and K Sato[1]

[1] Department of Micro-Nanosystem Engineering, Nagoya University, Nagoya 464-8603, Japan
[2] Laboratory of Physics, Helsinki University of Technology, POB 1100, 02015 HUT, Finland
[3] Department of Physics, Nagoya University, Nagoya 464-8602, Japan

E-mail: mag@kaz.mech.nagoya-u.ac.jp

Received 13 September 2006, in final form 17 January 2007
Published 20 March 2007
Online at stacks.iop.org/JMM/17/S27

Abstract
We propose that the formation of zigzag structures on Si(1 1 0) during anisotropic etching is mainly a result of the formation of inhomogeneous regions in the etchant due to diffusion phenomena. In the same way as the presence of these etchant inhomogeneities results in step bunches on miscut (1 1 1) surfaces, it results in zigzags on the (1 1 0) surface. To support this proposal, we present an incremental activity monitoring (IAM) method for the simulation of step bunching using a kinetic Monte Carlo scheme. For stepped (1 1 1) surfaces, comparison with a previous step density monitoring (SDM) method shows that IAM is typically faster by one order of magnitude and is well suited for the simulation of step bunching. By applying IAM to (1 1 0), the formation of zigzag structures can be simulated, strongly suggesting that the morphology of this surface is dominated by the formation of inhomogeneous regions close to the surface in the etchant phase.

(Some figures in this article are in colour only in the electronic version)

1. Introduction

Anisotropic wet chemical etching of crystalline silicon is a widely popular process for the fabrication of microstructures, where it is used alone or in combination with other techniques. From an engineering point of view, device performance has become dependent on the quality of the micromachined surfaces, in particular on their morphology and roughness. From a fundamental perspective, there remains interest in understanding a number of morphologic features. As an example, there has not been a satisfactory explanation for the surface morphology of anisotropically etched Si(1 1 0), which involves the formation of zigzag structures [1, 2], as shown in figure 1. It has been proposed that the zigzags appear because they minimize the surface free energy [4]: the crystallography of Si(1 1 0) allows the removal of a full row of atoms and its placement somewhere else on the surface with no energy cost and a gain in entropy. Perhaps the most accepted explanation assumes the presence of micromasking so that some of the atom rows remain temporarily frozen, thus becoming the zigzag peaks, while other rows are removed, thus becoming the zigzag valleys [5, 6]. Usually, the zigzags are analyzed in conjunction with the formation of nosed structures, also typically explained as the result of micromasking [5–7]. In this study we propose that the morphology of (1 1 0) is strongly controlled by diffusion phenomena which lead to the formation of inhomogeneous regions in the etchant, i.e. locally supersaturated and/or undersaturated domains. Just as these inhomogeneities generate step bunches on miscut (1 1 1) surfaces, they produce zigzag structures on (1 1 0). The paper focuses on the presentation of a faster method for the simulation of step bunching using a kinetic Monte Carlo (KMC) approach. Our simulations of the surface morphology and its time evolution for stepped (1 1 1) and (1 1 0) strongly suggest that the zigzag structures are another manifestation of the presence of diffusion inhomogeneities.

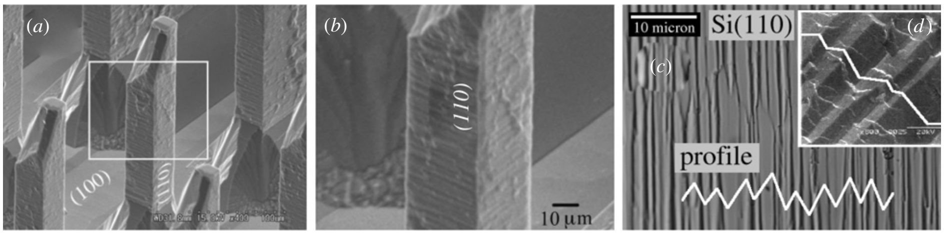

Figure 1. Impact of surface morphology on engineering applications: formation of zigzag structures on (1 1 0) facets during anisotropic etching of Si. (*a*) Micro-fabrication of micro-needles on a Si(1 0 0) wafer [3]. (*b*) Detail of (1 1 0) facets, region highlighted in (*a*). (*c*) Detail of (1 1 0) morphology showing the zigzag profile. (*d*) 3D view of the zigzag structures.

In this paper, the terms *diffusion* and *diffusion phenomena* are used to refer to the transport of the reactants and/or products to/from the locations where they are being consumed/produced. This transport takes a characteristic time τ, which is typically larger than the typical time δt between the atomistic, surface removal reactions ($\tau \gg \delta t$). This transport delay and the existence of intense activity at the steps (in comparison to the rather inert terraces) results into a lower etchant concentration at the steps, even in the presence of a steady flow of reactants into such regions in order to neutralize their continuous consumption. In this way, etchant depletion regions (or inhomogeneities) are formed in the vicinity of the steps.

The formation of these etchant inhomogeneities on stepped Si(1 1 1) surfaces has been recently investigated experimentally and computationally by Garcia *et al* [8, 9]. They concluded that the existence of inhomogeneities can satisfactorily explain the formation of step bunches on stepped (1 1 1) and thus the overall surface morphology of the (1 1 1) surfaces. Additionally, experimental work by Tan *et al* has previously stressed the importance of diffusion phenomena in order to explain the variations in the etch rate and activation energy of (1 1 1) and other vicinal orientations due to the proximity of the masking patterns [10]. In this paper, we explore computationally whether step bunching due to etchant inhomogeneities can explain the formation of the zigzag structures on Si(1 1 0). With this target in mind, we have developed an alternative method for the simulation of step bunching.

The paper does not focus on the formulation of a physical model of the diffusion phenomena, nor on the determination of the key diffusing species, which in principle can be any of H_2O, OH^-, the etchant cations (such as K^+ in KOH or TMA^+ in TMAH) and the reaction products (such as $Si(OH)_4$). We simply assume the presence of inhomogeneous regions in the etchant and explore what implications this can have on the morphology of (1 1 0).

2. Including diffusion phenomena in a kinetic Monte Carlo simulation: step density monitoring (SDM)

We are interested in including the effects of diffusion on the surface morphology in a simulation of anisotropic etching that uses the kinetic Monte Carlo (KMC) method. In this method, a surface atom i gets a removal rate k_i^0 whose value depends on the actual configuration of the neighborhood. For instance, k_i^0 can depend on the number of first and second neighbors, n_i^1 and n_i^2, as in $k_i^0 = k^0(n_i^1, n_i^2)$ [11]. Alternatively, as in most etching models, only a few distinct surface sites might be considered (such as the terrace and step monohydrides, the vertical and horizontal step dihydrides, etc [12]) and the number of different k_i^0s reduces to only 5–10.

Following Garcia *et al* [8, 9], we assume that the existence of diffusion phenomena during anisotropic etching results in the formation of inhomogeineities. Accordingly, one expects a decrease in the reaction rates of the atoms affected by depletion regions. Similarly, an increase in the rates is expected if the reaction is strongly exothermic and the local rise in the temperature increases the rates more than the etchant depletion reduces them. The situation can be mathematically described by introducing a scalar field D (referred to as the *diffusion factor*) such that $D = 1$ denotes that the etchant is in its normal, homogeneous state, $D < 1$ describes the presence of a depletion inhomogeneity and $D > 1$ describes the occurrence of a local increase in the etch rate:

$$k_i = D(\mathbf{r}_i)k_i^0. \tag{1}$$

Here, k_i is the atom removal rate under the effect of an inhomogeneity. Equation (1) stresses the fact that the diffusion factor $D = D(\mathbf{r})$ takes values that depend on the position $\mathbf{r}$ on the surface. Although not explicitly written, D will also change with time. As a result, also the rates k_i typically depend on time.

Since the formation of the inhomogeneities correlates with the activity on the surface and the activity is typically concentrated at the steps, Garcia *et al* assume that the diffusion factor D is proportional to the step density ρ, as considered in kinematic wave theory [9]:

$$D(\mathbf{r}) = 1 + a\rho(\mathbf{r}). \tag{2}$$

Here, the step density ρ is in the range [0, 1] and a is a model parameter. If $a = 0$, no diffusion effects are simulated as the etchant is completely homogeneous. In an experiment, this is equivalent to vigorous stirring conditions. If $a > 0$, the diffusion phenomena result in 'boosting' of the reaction rates (also referred to as 'acceleration') and, if $a < 0$, diffusion results in a 'slow-down' of the rates (i.e. 'deceleration'). In this case one must take care that the diffusion factor D never becomes negative or zero (e.g. by using the constrain that $a > -1$). As explained below, both $a > 0$ and $a < 0$ are expected to produce step bunching. The two situations should correspond to experiments where stirring is not used.

Within this formulation, the problem of simulating diffusion effects (such as the formation of etchant

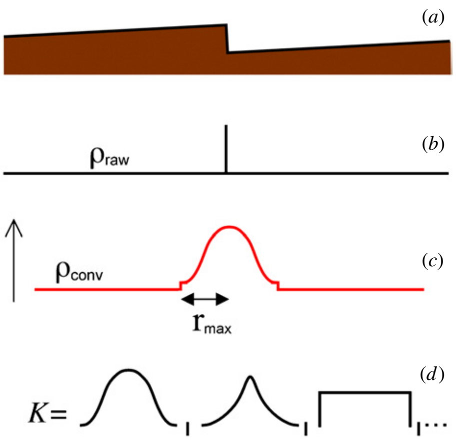

Figure 2. Relation between the convoluted and the raw step densities for a simple case (only one step). (*a*) A step between two terraces. (*b*) Definition of the raw step density ρ_{raw}. (*c*) The convoluted step density ρ_{conv}. (*d*) Different shapes for the convolution kernel.

inhomogeneities and step bunching) is reduced to a large extent to that of monitoring the step density ρ. Thus, we refer to this approach as step density monitoring (SDM).

2.1. The convolution method

In order to calculate the diffusion factor D, the step density ρ needs to be defined and monitored. Garcia *et al* have used a convolution method which can be described as follows. Let us consider *raw* and *convoluted* step densities (ρ_{raw} and ρ_{conv}, respectively). For the raw step density, the terrace sites are assigned the value '0' and the non-terrace sites (typically step sites) are assigned '1', as shown schematically in figures 2(*a*)–(*b*). Mathematically, the raw step density is defined as

$$\rho_{\text{raw}}(\mathbf{r}) = \sum_i \delta(\mathbf{r} - \mathbf{r}_i). \tag{3}$$

The convoluted density ρ_{conv} results from summing the values of the so-called convolution kernel (K) to a region of radial extent $r_{\max}$ centered at every non-terrace site (i.e. centered at the ones of the raw step density), as indicated in figure 2(*c*). This can be regarded as a stamping process that uses the kernel function values in the $r_{\max}$ region as the stamp. As shown in figure 2(*d*), different convolution kernels (or *blur functions*) can be used such as an exponential decay, a Gaussian or a hat function, to name only some examples. We will refer to the $r_{\max}$ region as the *blur neighborhood*. Mathematically, the convoluted step density can be described as

$$\rho_{\text{conv}}(\mathbf{r}) = \int K(\mathbf{r} - \mathbf{r}')\rho_{\text{raw}}(\mathbf{r}')d\mathbf{r}' = \sum_i K(\mathbf{r} - \mathbf{r}_i). \tag{4}$$

The convoluted step density gets the highest value along the steps and decays continuously as we move away from a step into the terrace area. ρ_{conv} looks as a blurred version of ρ_{raw}, as schematically shown in figure 2.

The convolution method consists in approximating the step density ρ in equation (2) by the convoluted step density ρ_{conv}. When two steps come close to each other as a result of fluctuations in their propagation velocities, the superposition of the two corresponding blur functions will lead to larger values of the convoluted step density in the intermediate region, thus enabling interaction between the steps. As shown in figures 3(*a*)–(*c*), the overlap between two blur functions creates a region where the diffusion factor $D(\mathbf{r}) = 1 + a\rho_{\text{conv}}(\mathbf{r})$ increasingly exceeds 1 (if $a > 0$), thus boosting the corresponding reaction rates (according to equation (1)). As the step pair propagates faster, it eventually catches up a third step, further increasing the value of D. As more steps are involved, big step bunches are formed (figure 3(*c*)). If $a < 0$ (figures 3(*d*)–(*f*)), D decreases when two steps come closer, thus slowing down the rates and making it possible for a third step to catch the pair, eventually also leading to step bunching.

In figure 3, the red plotted curves in (*a*)–(*c*) represent the onset of inhomogeneous regions in the temperature of the etchant, locally leading to acceleration of the corresponding step or step-bunch. The blue curves for (*d*)–(*f*) correspond to the formation of inhomogeneities in the concentration of the etchant, which will in principle lead to deceleration. Note, however, that etchant depletion does not necessarily always lead to deceleration: it is known that the etch rate can increase by decreasing the concentration, especially in the low concentration range. Thus, the red curves in (*a*)–(*c*) might as well describe the case of etchant depletion. Even if the exact link between acceleration/deceleration and the temperature/concentration inhomogeneities remains imprecise, the general argument is still solid. The existence of acceleration and/or deceleration of the step velocities can be correlated to the formation of inhomogeneities, and is ultimately due to the delay between diffusion transport and surface reactivity.

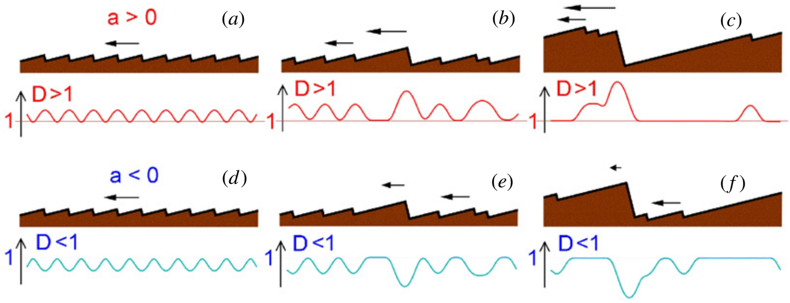

Figure 3. Formation of step bunches due to the presence of inhomogeineities: (*a*)–(*c*) acceleration; (*d*)–(*f*) deceleration. The extent of the horizontal arrows describes the magnitude of the step velocity.

From a computational perspective, the central idea in order to generate bunching is to encourage the overlap between active regions, which typically correspond to the steps. The convolution method directly follows the position of the steps and spreads them out by means of the convolution operation, thus enabling overlapping.

2.2. Difficulties of the convolution method

There are two main difficulties in using the convolution method:

(i) *Time-dependent rates*. Since the diffusion factor D can take many different values and change with time at every surface atom, the reaction rates k_i after multiplication by D (see equation (1)) constitute a large set of time-dependent rates. This occurs even if the number of the original rates k_i^0 is limited to just a few. In order to use variable rates in a kinetic Monte Carlo simulation, an appropriate method must be used to ensure that the most probable event is selected at each time step. For that purpose, the most general method that can be used is the K-level search (KLS) [13]. KLS is a general tree search method that includes many particular cases such as the binary search and the linear search. In [9] Garcia *et al* present a 'hybrid' method based on an N-fold search [14] of the next event out of the small set of time-independent k_i^0 rates followed by the acceptance (or rejection) of the event by comparing a random number to the value of the diffusion factor D_i (Metropolis acceptance/rejection criterion [15]). Although the approach is completely correct, the use of the Metropolis criterion can lead to slow computations due to frequent rejection, specially in the case of deceleration, where D is typically smaller than 1. An efficient manner to deal with time-dependent rates in a KMC simulation is the use of the KLS method in one of its most efficient variants (e.g. binary search and other choices). Our kinetic Monte Carlo simulation program (which we refer to as TAPAS: three-dimensional anisotropic processing at all scales) uses the KLS method.

(ii) *Slow computations*. The use of a convoluted step density (ρ_{conv}) requires defining a *typically large* blur neighborhood of size $r_{\max}$ for every atom (at least in the crystallographic basis). Here, 'typically large' means about 2 nm in radius, which amounts to a volume with more than a thousand Si atoms, corresponding to several hundreds of atoms on the surface. Besides the extra use of memory to keep these blur neighbors, updating the values of ρ_{conv} to reflect each event during the system evolution requires a loop over several hundreds of blur neighbors, significantly slowing down the performance of a kinetic Monte Carlo simulation. According to our implementation of the convolution method, the computing time per event is typically increased by an order of magnitude as compared to a simulation that does not use the blur neighborhood (see section 4.1). Thus, if possible, one would like to reduce the size of $r_{\max}$ as much as possible. A way to do this is presented in section 3.

3. Incremental activity monitoring (IAM)

Perhaps the most basic idea in order to simulate step bunching is to enable the overlap between active regions, which are typically the steps. The convolution method records the positions of the steps and spreads them out using the convolution operation, thus increasing the chances for overlapping. An alternative manner to enable overlapping is to keep track of the past positions of the steps so that overlapping can occur when a step enters a previously marked region. This approach assumes that the diffusion processes (i.e. the transport of reactants and/or products in the etchant phase close to the silicon surface) are typically slower than the actual atom removals at the surface. As a result, the etchant builds up a memory of the activity in the past. In a way, the convolution method indirectly states that the diffusion inhomogeneities follow the motion of the steps instantaneously and that they are symmetric ahead and behind the steps. But it might be more realistic to consider that the inhomogeneities are *asymmetric*, lagging more behind the steps. If this is the case, recording the surface activity in the past is a simple yet powerful manner of introducing step bunching in a KMC simulation, as explained below.

All that is required is to define an integer variable, the *incremental activity* (A), whose value is incremented by one unit at the first and second neighbors of each removed atom. This provides a way to record the past activity, i.e. the past removals. As more incremental updates are realized into A, a moving step leaves on A a track of its previous locations. In other words, the surface atoms keep a memory of when their neighboring atoms have been removed. The larger A is, the closer in time is the previous removal. The incremental activity monitoring (IAM) method consists in replacing the step density ρ in equation (2) by the *normalized activity* defined as $\rho_{\text{A}} = A/\max(A)$.

In order to prevent an infinite memory, A should be refreshed periodically. We find that truncation of the tails works satisfactorily. Truncation means periodically finding the minimum value $A_{\min}$ and updating as $A = A - A_{\min} - \Delta A$, as shown in figure 4. ΔA is a small value (e.g. $\Delta A = 1$) used to ensure truncation even if $A_{\min} = 0$. If the refresh rate is too short, A follows the steps closely but only weak overlapping can be realized. If the refresh rate is too large, we keep too much memory of the past activity. By using an intermediate refresh rate, realistic results can be obtained. Thus, we define a truncation interval ΔT (the inverse of the refresh rate) that allows control of the amount of past memory in the etchant. We measure ΔT in units of removal events, expressed as a fraction with respect to the number of particles on the surface. For instance, $\Delta T = 0.20$ means that truncation occurs every time that 20% of the surface atoms have been removed.

We have noticed that additional truncation of the top values improves the results obtained by the previous tail truncation. This head truncation does not need to be performed periodically. Since A is continuously incremented, it is very easy to keep track of the maximum value and thus limit the values of A not to exceed a predefined maximum $A_{\max}$.

In principle, the normalized activity ρ_{A} can alternatively be periodically recalculated from scratch, e.g., as the

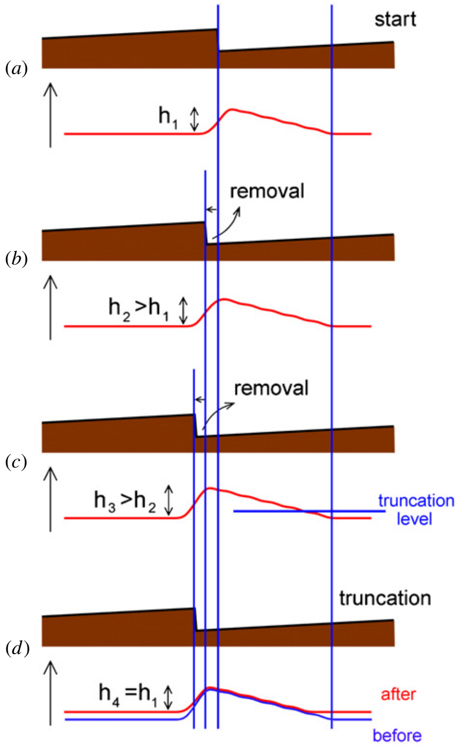

Figure 4. Schematic representation of the truncation method.

convolution of the raw step density ρ_{raw} by using any of the convolution kernels of figure 2(*d*) over a region of extent $r_{\max}$. From the point of view of the convolution method, this would ensure that the normalized activity is 'correctly' calculated as the step density periodically (i.e. $\rho_{\mathrm{A}} = \rho_{\mathrm{conv}}$ at the beginning of each period). However, this also destroys the memory of recent past diffusion processes. If the inhomogeneities in the etchant should be lagging behind the activity on the surface, this recalculation completely destroys the recent shape of the inhomogeneities. Besides, this approach can involve computationally expensive loops if $r_{\max}$ is large, making it slower, whilst the truncation method does not involve any blur neighborhood loops, making it faster. Thus, we favor the use of the truncation method against recalculation. The main implementation headlines of the incremental activity monitoring method using truncation are given in appendix A.

4. Results

In this section we compare the results obtained using our implementation of the previously existing step density monitoring method and the newly proposed incremental activity monitoring method. Focus is first placed on the relative efficiency of the two methods and the ability to simulate step bunching. At the end of this section we provide simulation results for Si(1 1 0), showing that the formation of zigzag structures on this orientation has the same physical origin as step bunching on miscut (1 1 1) surfaces.

4.1. Comparison between SDM and IAM

In order to perform the comparison tests between the SDM and IAM methods, we consider miscut Si(1 1 1) surfaces of type $(h\,h\,h \pm 2)$, where h is an integer, similar to those used by Garcia *et al* [8, 9]. Figure 5 shows two examples of miscut surfaces.

For improved quantitative comparison of the IAM and SDM methods, we measure the terrace width distribution (TWD) at different times, including early and late moments, as shown in figure 6. The TWD is constructed by analyzing a number of 10 to 50 cross sections of the surface, as indicated in figure 6(*a*). For every cross section, a histogram is obtained for the different terrace widths found. By gathering all the histograms together and ordering the results according to increasing terrace width we obtain the terrace width distribution.

The TWD is typically Gaussian at early times (figure 6(*b*)) and becomes exponential as the steps bunch together, which results in a small number of wide and a large number of narrow terraces (figure 6(*c*)). As it turns out, it is not necessary to look at the TWD plots nor at the surfaces themselves in order to know if step bunching occurs and, if so, how strongly. The TWD can be characterized by one single number that contains this information. We have found that the variance of the TWD and the maximum terrace width are both good measures

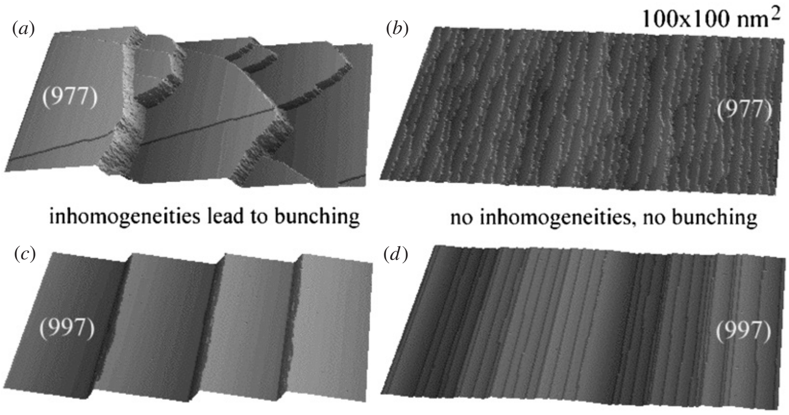
(a)
(977)
(b)
100x100 nm2
(977)
inhomogeneities lead to bunching
no inhomogeneities, no bunching
(c)
(997)
(d)
(997)

Figure 5. Examples of simulated $(hhh \pm 2)$ surfaces. Step bunching generated using the IAM method ($a = 10.0$, $\Delta T = 0.4$, $A_{\max} = 18$, $\Delta A = 1$). No bunching obtained with $a = 0$.

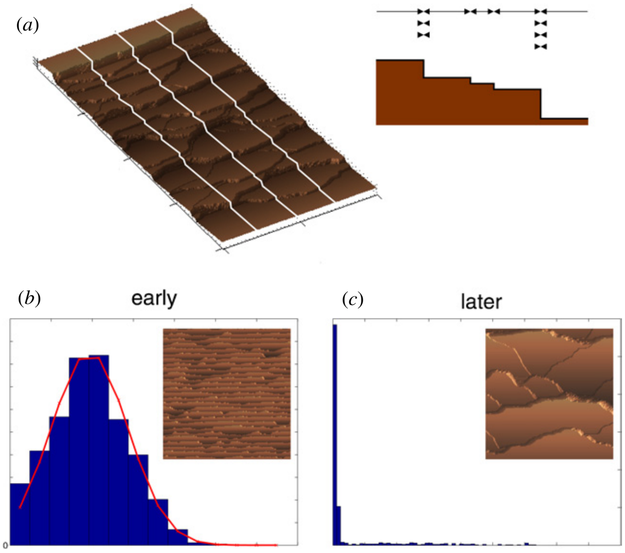

Figure 6. Construction of the terrace width distribution (TWD) for monitoring step bunching. (*a*) Definition of the terrace width histogram used to obtain the TWD. (*b*)–(*c*) TWD for early and late times during the same simulation.

in order to characterize step bunching. The terrace width variance corresponds to the second moment of the terrace width distribution.

Figure 7(*a*) shows the standard deviation of the terrace width (i.e. the square root of the variance) and the maximum terrace width as functions of the number of removal events in the simulations. For both the IAM and SDM methods, step bunching develops with increasing time as the number of simulated events increases. At early times, the TW variance and maximum are small, signaling the absence of bunches. At longer times, both the TW variance and maximum increase to a maximum value, implying the existence of stable step bunches, as supported by figure 7(*b*). Even though both methods produce bunching, SDM generates bunches only moderately whilst IAM exhibits large, well-packed, bunches that are also more realistic. In general, we find that it is actually rather difficult to produce strong bunching with our implementation of the SDM method. The bunching results by Garcia *et al* can be placed somewhere in between our SDM and IAM implementations: IAM gives the strongest bunching, followed by [9] and finally SDM gives the weakest features. The use of a hat function for the convolution kernel instead

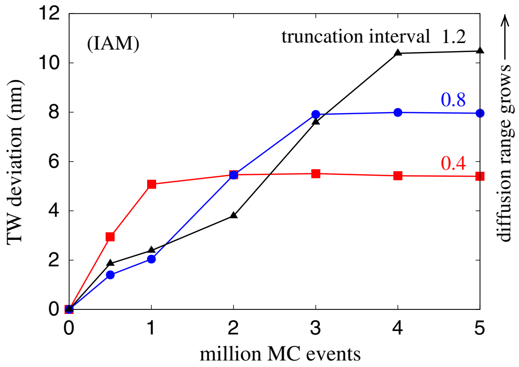

Figure 8. Dependence of the maximum bunch size on the truncation interval ΔT for the IAM method.

of a Gaussian or an exponential provides slightly improved bunching with SDM, but never as strong as with IAM.

The weaker bunching obtained with SDM as compared to the results from Gracia *et al* is not due to differences in the kinetic Monte Carlo implementation. We use the same program for both the IAM and SDM methods and we find strong bunching with IAM. This includes the same K-level search algorithm for finding the next event in the kinetic Monte Carlo simulations. We have independently tested the 'hybrid' kinetic Monte Carlo implementation by Garcia *et al* in other systems and it always produces the correct time evolution. As a matter of fact, the hybrid method can be shown to be equivalent to any K-level search, as shown in appendix B. We presume that the differences in bunching might be related to a different choice of removal rates, resulting into quite isotropic step flow on (1 1 1) in their case and quite anisotropic in our case. For the more isotropic case, one would expect that the inhomogeneities become more round and thus overlapping may improve.

The size of the largest bunch can be easily controlled in the IAM method. Increasing the absolute value of parameter a will produce larger bunches. Also increasing the truncation interval ΔT and/or the maximum activity value $A_{\max}$ will produce the same effect. As an example, figure 8 shows that the size of the bunches increases as the truncation interval ΔT is increased. This is expected since enlarging ΔT corresponds to increasing the amount of memory of past activity kept in A. For a fixed truncation interval, figure 8 shows that the size of the bunches increases linearly with time (the number of events in that figure) and finally approaches an asymptotic value,

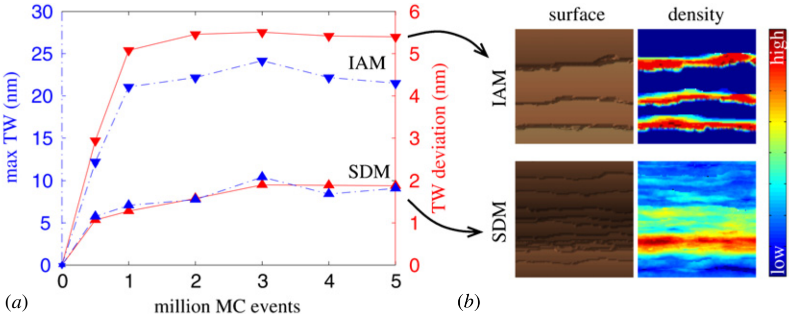

Figure 7. Comparison of step bunching characteristics produced by the IAM and SDM methods. IAM produces larger, clearer bunches.

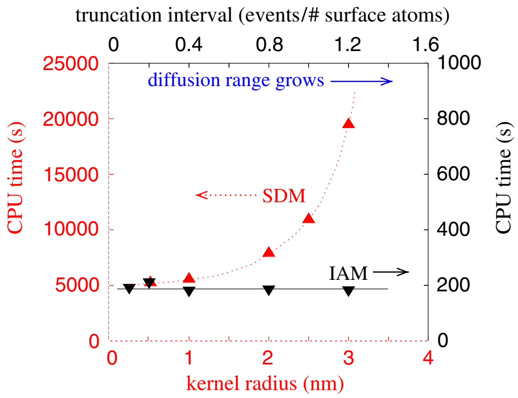

Figure 9. Comparison of computational efficiency of the IAM and SDM methods. IAM is typically an order of magnitude faster than SDM. Top and right axes for IAM. Bottom and left axes for SDM. Size of simulations: 50 × 50 nm^2.

corresponding to the typical size of stable bunches. As the truncation interval is increased, larger bunches are produced, leading to surfaces with larger roughness values, i.e. larger interface widths. For the case of an infinite truncation interval (which corresponds to never realizing the truncation operation) one single, large bunch is developed always independently of the size of the system.

For the case of the IAM simulation shown in figure 7, the parameter values were $a = 10.0$, $\Delta T = 0.4$, $A_{\max} = 18$ and $\Delta A = 1$, and for the SDM simulation, $a = 10.0$ and a hat function as the kernel with $r_{\max} = 2.0$ nm. The size of the simulations was 50 × 100 nm^2, of which a region of approximately 30 × 30 nm^2 is shown. Similar results where obtained with larger sizes (up to 300 × 300 nm^2).

Figure 9 shows a comparison of the computational efficiency of the IAM and SDM methods for similar choices of parameters. The total CPU time spent to complete 50 million removal events is plotted as a function of the kernel radius for SDM (section 2.1). The convolution kernel was a hat function in all cases. For IAM, the same CPU time is plotted against the truncation interval. The size of the inhomogeneities (denoted as the 'diffusion range' in the figure) increases as both the kernel radius and the truncation interval increase. The figure shows that the computational cost of the IAM method is independent of the diffusion range whilst increasing the size of the bunches rapidly becomes prohibitive for the SDM method. In addition, IAM is at least one order of magnitude faster than SDM. For instance, IAM is about 25 times faster for the case shown in figure 7 ($\Delta T = 0.4$) and it can be hundreds of times faster if larger inhomogeneities are simulated. As explained in section 2.2, updating the convoluted density in SDM is a heavy, time-consuming task. Note that the two x-axes in figure 9 are not completely equivalent, meaning that the diffusion ranges in the two methods do not exactly match. However, this is immaterial since the IAM curve is completely horizontal.

In conclusion, the newly proposed incremental activity monitoring method provides faster simulations and improved step bunching characteristics as compared to our implementation of the convolution-based step density monitoring method.

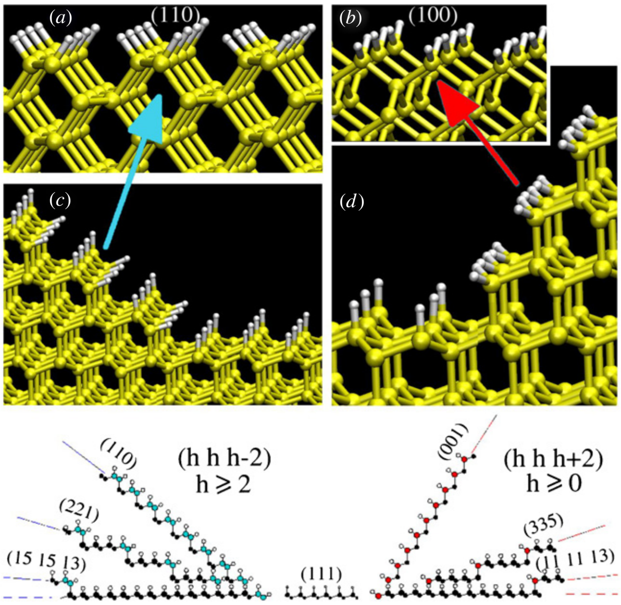

Figure 10. (1 1 0) and (1 0 0) as limiting examples of stepped (1 1 1) surfaces.

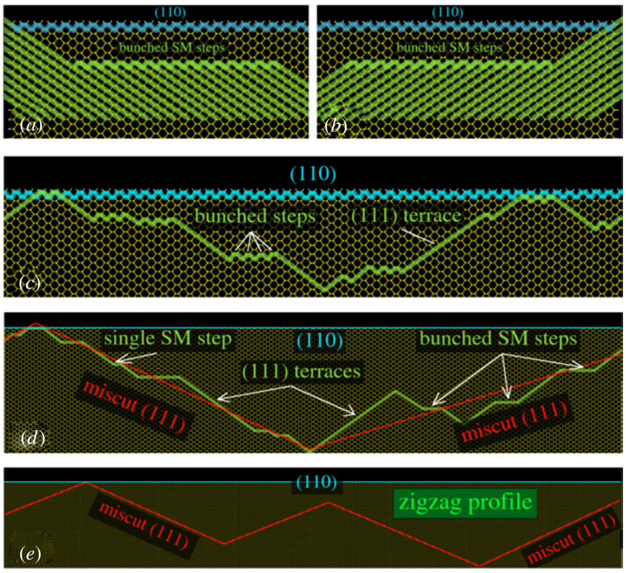

Figure 11. Schematic formation of zigzag structures on (1 1 0).

4.2. *Morphology of Si(1 1 0)*

As noted previously, the main objective of our study is to describe the formation of zigzag structures on Si(1 1 0). In this section we describe how the zigzags can appear as a result of the formation of etchant inhomogeneities.

The (1 1 0) orientation can be considered as a limit example of a stepped (1 1 1) surface, in this case without terraces, as shown in figure 10. Thus, the ideal (1 1 0) surface can be regarded as a large bunch of steps, as outlined in figures 11(*a*)–(*b*) for the two equivalent terrace families. From this perspective, the over sized bunch will necessarily split into smaller bunches during etching as a result of the same step velocity fluctuations which generate step bunching in misoriented (1 1 1) surfaces. This is graphically explained

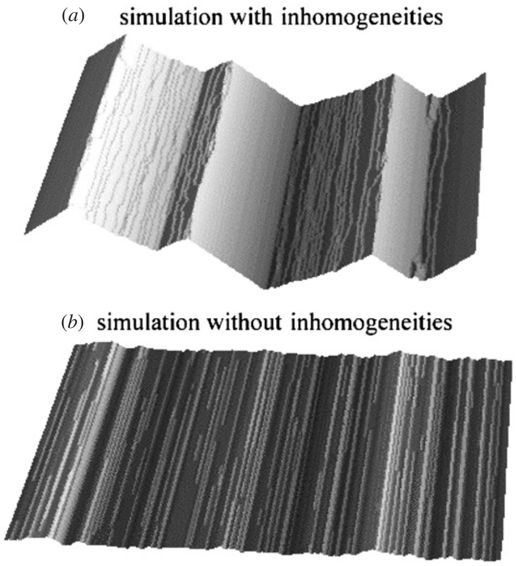
(*a*) simulation with inhomogeneities
(*b*) simulation without inhomogeneities

Figure 12. Formation of zigzag structures on (1 1 0). (*a*) Simulation using IAM ($a = 2.0$, $\Delta T = 1.4$, $A_{\max} = 18$, $\Delta A = 1$). (*b*) Simulation without inhomogeneities ($a = 0$). The system size is $100 \times 100\ \text{nm}^2$. Compare figure 1(*c*)–(*d*).

in figure 11(*c*), where the initial (1 1 0) surface transforms into a collection of (1 1 1) terraces separated by bunched SM steps. Considering larger systems, this behavior leads to a zigzag profile where each segment is in reality a collection of (1 1 1) terraces separated by step bunches, as described in figures 11(*d*)–(*e*).

In order to test the previous qualitative explanation we have simulated anisotropic etching on (1 1 0) using the IAM method. Figure 12 shows typical simulated morphologies that can be compared with a typical experiment, such as figures 1(*c*)–(*d*). It becomes apparent that the presence of etchant inhomogeneities, which leads to step bunching on miscut (1 1 1) surfaces, results in zigzags on (1 1 0). Thus, we conclude that the zigzags on (1 1 0) appear as a result of diffusion transport.

5. Additional methods

During our study of zigzag formation on (1 1 0), we also explored other alternative approaches for the incorporation of inhomogeneities and the simulation of step bunching. Although we consider IAM as the best alternative from a computational perspective, we report here on two additional methods in the hope that some readers might find some of the ideas stimulating.

5.1. Timed diffusion (TD)

In this method, the use of time management is the main difference from the other methods. We simply monitor the surface activity and use time stamps to keep memory of which regions have been active and when. The main features of the method are as follows: (a) when an atom is removed from the surface, local acceleration or deceleration due to diffusional transport is assumed to affect the first and second neighbors; (b) the current time is recorded as a time stamp on the first and second neighbors, enabling boosting or slow-down of their rates for a period of time τ (starting at the current time); (c) if an atom is stamped again (due to some other neighbor removal) the diffusion time will be extended (i.e. by adding τ to the remaining time) and the etching rate will be changed accordingly. These features are realized by using the following diffusion factor:

$$D = D(t) = 1 + aT(t). \tag{5}$$

$T(t)$ is a time decay function such as a hat function of duration τ, a decaying exponential with decay time τ or some other suitable choice, as shown in figure 13. As expected, D boosts the etching rates if $a > 0$ and slows them down if $a < 0$.

This method produces step bunching by considering a small neighborhood (only first and second neighbors) in comparison to the step density monitoring method (typically involving blur neighborhoods with hundreds of atoms). A disadvantage is that the method has an additional computational cost for updating the time stamp and τ values for every process after each atom removal.

Timed diffusion is close in spirit to incremental activity monitoring, at least in the sense that both methods monitor and keep a record of the active regions of the surface, which enables overlapping. From a computational point of view, the use of time stamps in TD makes the program more complicated and slower, and requires a larger use of memory.

5.2. Height dependent diffusion (HDD)

An alternative method is based on the observation that zigzag formation on (1 1 0) will occur if some of the steps are essentially frozen, thus becoming the zigzag peaks, whilst other steps are essentially boosted, thus becoming the zigzag valleys. This can be realized by using a diffusion factor D that depends on the relative height, such as the following:

$$D(\mathbf{r}_i) = 1 + a\,\mathrm{e}^{-\frac{h_i - h_{\text{ave}}}{w}}. \tag{6}$$

Here, h_i is the height of the surface atom situated at position $\mathbf{r}_i$, $h_{\text{ave}} = \sum_{i=1}^{N} h_i/N$ is the average height of the surface and $w = \left[\sum_{i=1}^{N}(h_i - h_{\text{ave}})^2/N\right]^{1/2}$ is the interface width. As schematically shown in figure 14, equation (6) decelerates

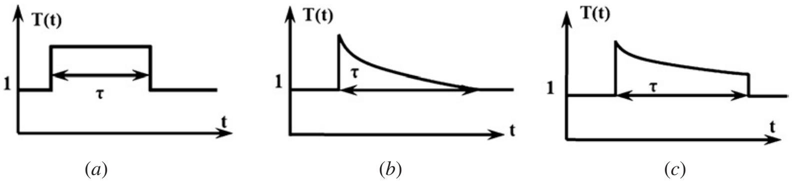

Figure 13. Examples of some time decay functions $T(t)$ for the timed diffusion method.

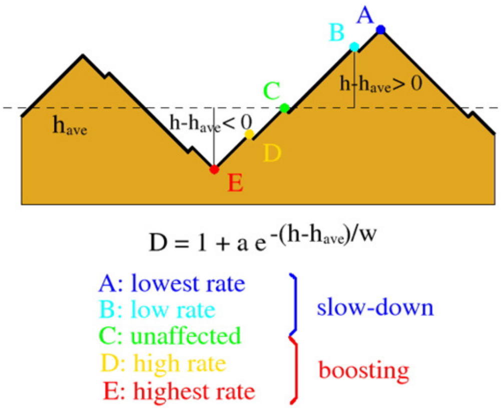

Figure 14. Examples of diffusion factor values for the height-dependent diffusion method.

steps A and B and accelerates D and E, while not affecting the rate of step C. When the surface is initially flat, statistical fluctuations produce small peaks and valleys, which then are amplified by the diffusion factor giving rise to the formation of the zigzag patterns.

The use of h_{ave} is not compulsory and other reference levels can be used, such as the minimum h_{min}, as in

$$D(\mathbf{r}_i) = 1 + a\,\mathrm{e}^{-\frac{h_i - h_{\text{min}}}{w^*}}, \tag{7}$$

where $w^* = h_{\text{max}} - h_{\text{min}}$. In this case, all atom rates are accelerated, the boosting factor being maximum for the zigzag valleys ($D = 1 + a$) and minimum for the zigzag peaks ($D \approx 1 + a/e$). Deceleration can be obtained by using, e.g., $D(\mathbf{r}_i) = a\,\mathrm{e}^{-\frac{h_i - h_{\text{min}}}{w^*}}$ with $0 < a < 1$.

Figure 15 shows an example of step bunching produced by this method. Comparison of frames (*a*) and (*b*) shows that the diffusion factor always reaches a peak at the bottom edge of a bunch. Note that the largest bunch (denoted with letter B in figure 15(*c*)) does not actually get the largest diffusion factor value. Instead, a smaller bunch (C in figure 15(*c*)) receives it. This behavior agrees with the fact that bunch C corresponds to the lowest region of the whole surface, so the diffusion factor should be largest according to equation (7).

The HDD method always leads to a large, single bunch independently of the system size. In this sense, the method is similar to IAM without truncation (see the discussion in relation to figure 8 in section 4.1). However, for HDD typically the step which eventually becomes the highest peak is already decided at an early stage, during the initial fluctuations, in the same way as the lowest valley typically corresponds to the first valley formed at the beginning. Although the formation of one single, final bunch in principle agrees with typical behavior found in growth, where the interface width often diverges with time, the step bunches during anisotropic etching typically reach limited proportions. Even though sometimes the peak-to-valley horizontal distance of the zigzags can reach the micron scale in the experiments, there is no evidence for a completely divergent behavior. Since a non-divergent behavior can be obtained by using IAM with a finite truncation interval and, additionally, we expect the largest inhomogeneities to be developed close to the largest step bunches (not beside the deepest features, as given by HDD), we perceive the morphologies produced with IAM as more realistic than those produced by the HDD method.

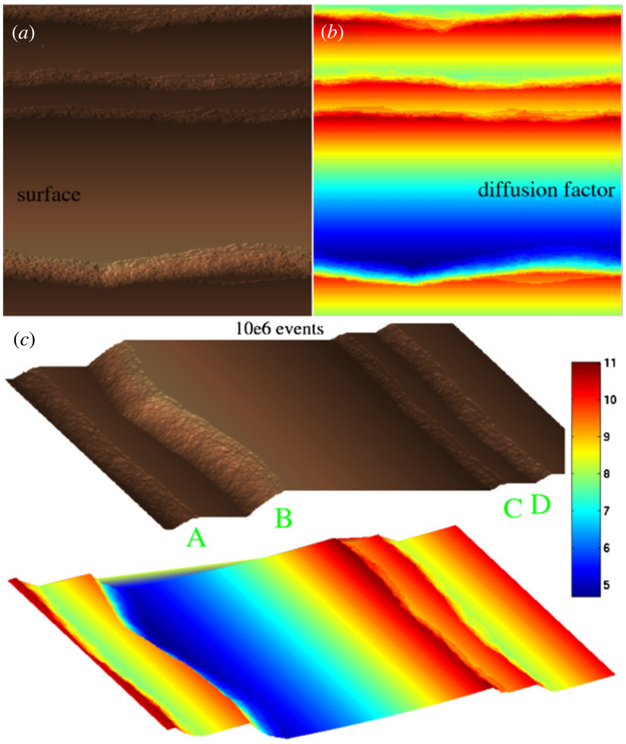

Figure 15. Time-shot from a simulation of step bunching on (977) using the height-dependent diffusion factor (equation (7), $a = 10.0$).

Another interesting feature of the HDD method is that the diffusion factor D essentially becomes a mirror image of the surface: D has a local/global maximum where the surface has a local/global minimum (see figure 15(*c*)). We have noticed a similar behavior when using the IAM method without truncation. In this case, the activity A increases continuously, effectively becoming a mirror copy of the surface itself. In this respect, the truncation used in IAM, which avoids the formation of this mirror behavior, can be considered as a way to introduce height differences in the HDD method which are local, e.g. involving a local height average h^l_{ave} or a minimum height h^l_{min} for small regions with characteristic size l. From this point of view, the IAM method can be understood as a HDD model where the height differences are local, instead of global as in the HDD method itself.

Since HDD is based on a global height reference level (i.e. h_{ave} or h_{min} or other choices), it is difficult to apply it to engineering applications where etch masks are used and a number of crystallographic orientations are likely to appear. HDD is only suitable for one surface at a time. If many surfaces appear in the system, the results will be quite unphysical.

6. Conclusions

We have proposed that the formation of zigzag structures on Si(1 1 0) during microstructure fabrication by anisotropic etching is mainly a result of diffusion phenomena. In brief, the explanation is as follows: (1 1 0), which can be considered as a stepped (1 1 1) surface, suffers step velocity fluctuations which lead to the formation of inhomogeneous regions in the etchant. The inhomogeneities in turn counter-affect the reaction rates of the steps, which leads to step bunching on (1 1 1) and zigzag formation on (1 1 0). To support this argument, we have introduced an incremental activity monitoring (IAM) method for the simulation of step bunching during anisotropic etching using a kinetic Monte Carlo scheme. Benchmark tests for stepped (1 1 1) surfaces against a previously existing step density monitoring (SDM) method show that IAM provides not only faster simulations but also an increased ability to reproduce bunching. As a result, we have been able to simulate the formation of zigzags on (1 1 0). This means that the morphology of (1 1 0) is dominated at least partly by the presence of etchant inhomogeneities.

Acknowledgments

The authors acknowledge the generous computer resources from the Center for Scientific Computing, Espoo, Finland. This research has been supported by the Finnish Academy of Science and Letters, Vilho, Yrjö and Kalle Väisälä Foundation, the Academy of Finland through its Center of Excellence Programme 2000-2011 and project SA 122603, the Japanese Center of Excellence Programme ‘Micro- and Nano-Mechatronics for Information-Based Society’ (2003-2007), the Finnish Cultural Foundation and the JSPS-Bilateral Program with the Academy of Finland.

Appendix A. Implementation of the IAM method

We present the main headlines of our implementation of the IAM method in a kinetic Monte Carlo simulation of anisotropic etching. In addition to the notation introduced in sections 2 and 3, we use the following acronyms:

TA	Target atom
FN	First neighbor
SN	Second neighbor
R	The total rate, which can change with time. Sum of the rates of all atoms currently on surface.
KLS	K-level search (see [13])
SAL	Surface atom list. List of atoms at the leaves of the KLS tree.

Pseudo code for IAM implementation

```
(i) Use KLS to find the next TA (to be removed).
    Let $i$ be the element in SAL corresponding to the chosen TA.
(ii) Increase time: $t = t + \delta t$, with $\delta t = 1/R$.
(iii) Remove chosen TA:
    (a) Update the rate $k_i$ stored at leaf $i$ in the KLS tree.
        (The old rate is replaced by a zero.)
    (b) Loop over the FNs and SNs of the TA:
        Let $j$ be the element in SAL corresponding to the currently considered FN or SN.
        (1) If FN was not in the surface, add it now. (Placing it at leaf $i$ prevents leaving holes in the KLS tree.)
            Give an initial value to the activity (e.g. copy value from TA):
                $A(\mathbf{r}_j) = A(\mathbf{r}_i)$.
        (2) If SN was not in the surface, do nothing.
        (3) If FN or SN was already in the surface:
            Increment the activity by one unit:
                IF($A(\mathbf{r}_j) < A_{\max}$) THEN
                    $A(\mathbf{r}_j) = A(\mathbf{r}_j) + 1$
                END IF.
            This is the key idea of the IAM method. It records the activity observed at TA into its neighborhood.
            Incrementing is limited by $A_{\max}$ in order to prevent $A$ from exploding.
        (4) Update $D$ for this FN or SN:
                if boosting ($a > 0$) use
                    $D(\mathbf{r}_j) = 1 + a\rho_{\mathrm{A}}(\mathbf{r}_j)$;
                if slowing-down ($a < 0$) use
                    $D(\mathbf{r}_j) = 1 + a(1 - \rho_{\mathrm{A}}(\mathbf{r}_j))$
            where $\rho_{\mathrm{A}}(\mathbf{r}_j) = A(\mathbf{r}_j)/A_{\max}$ is the normalized activity, as defined in section 3.
        (5) Calculate the new rate using the diffusion factor:
                $k_j = D(\mathbf{r}_j)k_j^0$.
        (6) Update the rate stored at leaf $j$ in the KLS tree.
            (Subtract the old rate and add the new one.)
(iv) Truncate the activity $A$ every $\Delta T$ events:
    Find the minimum of $A$ and update $A$ as:
        $A = A - A_{\min} - \Delta A$
    where $\Delta A = 1$ is used to ensure truncation even if
    $A_{\min} = 0$. Update $D$, $k$ and the KLS tree correspondingly
    (as in (iii.b)4 through (iii.b)6, but this time always using
        $D(\mathbf{r}_j) = 1 + a\rho_{\mathrm{A}}(\mathbf{r}_j)$
    for both $a > 0$ and $a < 0$).
(v) Update R.
    Calculate averages of observables (global etch rate, etc . . . ) and write output if needed.
(vi) Go back to first item (i).
```

Note: the diffusion factor is updated according to equation (2), except in (iii.b) if $a < 1$ (deceleration), where the expression $D = 1 + a(1 - \rho)$ is used. $(1 - \rho)$ predicts the activity ahead of the steps in the same manner as ρ records the activity behind them.

Appendix B. Is the ‘hybrid’ method correct?

According to equation (1), the removal rate k in the presence of inhomogeneities is the product of the diffusion factor D and the ‘raw’ rate k^0. In [9] Garcia *et al* present a ‘hybrid’ method to find the next removed atom. This is a two-stage approach including (i) an N-fold search [14] in the time-independent set of k^0 rates, consisting in grouping the k^0 rates (also referred to as events) into bins, choosing a bin and then picking up randomly an event in the bin; and (ii) accept (or reject) the chosen event by comparing a random number to the value of the diffusion factor D (Metropolis acceptance/rejection criterion [15]). As shown in [16], the N-fold method in (i), also known as BKL, is a particular example of a more general binning method [17] and can be replaced by many different, tree-based, search methods, such as a binary search (typically faster than

BKL) or a linear search (typically slower than BKL), which are two particular examples of the most general tree search method, known as K-level search (KLS) [13]. In this sense, (i) can be simply regarded as 'a search' which can be carried out in practice in multiple manners (binary, linear, KLS, binning and BKL methods). Here we are interested in the question of whether combining 'a search' in the k^0 set of rates in (i) with a Metropolis acceptance according to D in (ii) is equivalent (or not) to 'a search' in the $k = Dk^0$ set of rates.

Let the system have M site types (indices $\alpha, \beta, \ldots$) with N_α sites each and 'raw' rates k^0_α. $N = \sum_\alpha N_\alpha$ is the total number of sites. Let, in addition, each site have a diffusion parameter D_i so that the probability for choosing the site is $p_i \propto D_i k^0_{\alpha_i}$, i.e. $p_i = C D_i k^0_{\alpha_i}$ where α_i is the type of site i. Normalization requires $\sum_i p_i = 1$ so $C = 1 \big/ \sum_i D_i k^0_{\alpha_i}$. The probabilities are then

$$p_i = \frac{D_i k^0_{\alpha_i}}{\sum_j D_j k^0_{\alpha_j}}. \tag{B.1}$$

Let us calculate the probabilities in the hybrid method. First, the BKL probability to choose a type α is

$$P^{\text{type}}_\alpha = \frac{N_\alpha k^0_\alpha}{\sum_\beta N_\beta k^0_\beta}. \tag{B.2}$$

The probability to pick site i is

$$P^{\text{site}}_i = \frac{P^{\text{type}}_{\alpha_i}}{N_{\alpha_i}} \tag{B.3}$$

$$= \frac{k^0_{\alpha_i}}{\sum_\beta N_\beta k^0_\beta}. \tag{B.4}$$

The hybrid probability to then accept the site for removal is

$$P^{\text{acc}}_i = C' D_i P^{\text{site}}_i \tag{B.5}$$

$$= C' \frac{D_i k^0_{\alpha_i}}{\sum_\beta N_\beta k^0_\beta}, \tag{B.6}$$

where $C'D_i$ is the probability to accept the removal at the second stage of the hybrid method (i.e. the Metropolis acceptance/rejection). It is possible that we choose to decline and nothing is done. Let the total probability for this be Q. In this case the procedure is repeated from the beginning. The total probability for the site i to be the one where the next removal will eventually happen is then

$$P^{\text{final}}_i = P^{\text{acc}}_i + Q P^{\text{acc}}_i + Q^2 P^{\text{acc}}_i + \cdots \tag{B.7}$$

$$= \sum_m Q^m P^{\text{acc}}_i \tag{B.8}$$

$$= \sum_m Q^m C' \frac{D_i k^0_{\alpha_i}}{\sum_\beta N_\beta k^0_\beta}, \tag{B.9}$$

i.e. the sum of the probability that acceptance will occur at the first trial plus the probability that it will be rejected and accepted at the second trial plus the probability that it will be rejected twice and then accepted and so on. Normalization now states $\sum_i P^{\text{final}}_i = 1$ which gives for the normalization constants $\sum_m Q^m C' = \sum_\beta N_\beta k^0_\beta \big/ \sum_i D_i k^0_{\alpha_i}$ yielding

$$P^{\text{final}}_i = \frac{D_i k_{\alpha_i}}{\sum_j D_j k^0_{\alpha_j}} = p_i. \tag{B.10}$$

So the removal probability in the hybrid method is correct.

References

[1] Shikida M, Masuda T, Uchikawa D and Sato K 2001 *Sensors Actuators* A **90** 223–31

[2] van Veenendaal E, Sato K, Shikida M and van Suchtelen J 2001 *Sensors Actuators* A **93** 219–31

[3] Shikida M, Ando M, Ishihara Y, Ando T, Sato K and Asaumi K 2006 *Sensors Actuators* A **125** 415–21

[4] Elwenspoek M and Jansen H 1998 *Silicon Micromachining* (Cambridge: Cambridge University Press) p 72–3

[5] Zubel I and Kramkowska M 2004 *Sensors Actuators* A **115** 549–56

[6] van Veenendaal E, Sato K, Shikida M, Nijdam A J and van Suchtelen J 2001 *Sensors Actuators* A **93** 232–42

[7] Gosálvez M A and Nieminen R M 2003 *New J. Phys.* **5** 100.1–100.28

[8] Garcia S P, Bao H and Hines M A 2004 *Phys. Rev. Lett.* **96** 166102

[9] Garcia S P, Bao H and Hines M A 2004 *J. Phys. Chem.* B **108** 6062–71

[10] Tan S, Boudreau R and Reed M 2001 *Sensors Mater.* **13** 303–13

[11] Gosálvez M A, Nieminen R M, Kilpinen P, Haimi E and Lindroos V 2001 *Appl. Surf. Sci.* **178** 7–26

[12] Kasparian J, Elwenspoek M and Allongue P 1997 *Surf. Sci.* **388** 50–62

[13] Blue J L, Beichl I and Sullivan F 1995 *Phys. Rev.* E **51** R867–R868

[14] Bortz A B, Kalos M H and Lebowitz J L 1975 *J. Comput. Phys.* **17** 10–8

[15] Metropolis N, Rosenbluth A W, Rosenbluth M N, Teller A H and Teller E 1953 *J. Chem. Phys.* **21** 1087–91

[16] Levi A C and Kotrla M 1997 *J. Phys.: Condens. Matter* **9** 299–344

[17] Maksym P A 1988 *Semicond. Sci. Technol.* **3** 594–6