

Учреждение российской академии наук
Институт прикладной математики им. М.В. Келдыша РАН

На правах рукописи

Карташев Всеволод Владимирович

**АЛГОРИТМИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
КОМПЛЕКСОВ ДЛЯ ЗОНДОВОЙ МИКРОСКОПИИ**

Специальность 05.13.11 –
математическое и программное обеспечение
вычислительных машин, комплексов и компьютерных сетей

Диссертация
на соискание ученой степени
кандидата физико-математических наук

Научный руководитель –
профессор, доктор физико-математических наук,
Платонов Александр Константинович

Москва – 2010

Оглавление

Введение.....	3
Глава 1. ОБЗОР ПРОГРАММНЫХ СРЕДСТВ ОБРАБОТКИ РЕЗУЛЬТАТОВ СКАНИРОВАНИЯ ПОВЕРХНОСТИ ТУННЕЛЬНЫМ МИКРОСКОПОМ...	7
§1. Зондовая микроскопия рельефа поверхности.....	7
§2. Программные пакеты для отображения измерений рельефа поверхности.	10
§3. Увеличение визуального разрешения нанорельефа.....	22
§4. Искажения, связанные с неопределённостью модели сканера.....	25
Выводы первой главы.....	32
Глава 2. АЛГОРИТМ ОПРЕДЕЛЕНИЯ ГЕОМЕТРИИ ОСТРИЯ ИГЛЫ.....	34
§1. Принцип работы туннельного микроскопа.....	35
§2. Алгоритм определения геометрии острия иглы туннельного микроскопа.	43
Выводы второй главы.....	46
Глава 3. ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО КОМПЛЕКСА ДЛЯ ИНТЕРПРЕТАЦИИ ДАННЫХ СКАНИРОВАНИЯ.....	47
§1. Форматы входных и выходных данных.....	47
§2. Особенности программной модели.....	58
§3. Проектирование комплекса с заменяемыми модулями.....	61
§4. Выбор языка программирования для реализации комплекса.....	66
§5. Модифицируемость программного кода.....	69
§6. Проектирование многопоточности.....	74
Выводы третьей главы.....	77
Глава 4. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ КОМПЛЕКСА ДЛЯ ОБРАБОТКИ ДАННЫХ СКАНИРОВАНИЯ.....	78
§1. Внутренний формат данных.....	78
§2. Органы управления и пользовательский интерфейс.....	81
§3. Задание туннельной оболочки острия иглы.....	84
§4. Средства оценивания точности определения рельефа.....	88
§5. Организация параллельных вычислений.....	89
§6. Структура программных средств реализованного комплекса.....	93
§7. Методика применения программного комплекса.....	97
§8. Средства настройки программного комплекса.....	109
§9. Использование комплекса в исследованиях рельефа поверхности.....	111
Выводы четвёртой главы.....	116
Заключение.....	117

Введение

Исследование поверхности с высоким пространственным разрешением является важным элементом разработки новейших технологий. В распоряжении исследователя в настоящее время имеется достаточно богатый арсенал приборов, которые обеспечивают получение изображения исследуемой поверхности с нанометровым разрешением. К ним относятся: просвечивающий и сканирующий электронный, ближнепольный оптический, атомно-силовой и туннельный зондовые микроскопы. Успехи электроники привели к тому, что в настоящее время из всех перечисленных приборов зондовая микроскопия является наиболее доступным методом исследований, обеспечивая, при этом, наивысшее разрешение – от ангстрема (у простых приборов) до сотых долей ангстрема (у прецизионных микроскопов). Экстремально малые размеры объектов, с которыми оперируют нанотехнологии, делают крайне затруднительной визуализацию объектов и процессов, протекающих с их участием. Следствием этого является проблема соответствия измеренного и реального состояний объекта наноизмерений.

В диссертации эта проблема рассматривается на примере средств обработки сигналов туннельной микроскопии, позволяющей получать изображения поверхности с атомарным разрешением. Интенсивное развитие туннельной микроскопии связано с успехами электроники, которые в 90-х годах сделали устройства зондовой нано- и микроскопии доступными для широкого применения.

В настоящее время парк приборов, обеспечивающих измерение геометрии нано- или микрорельефа в поле зрения микроскопа, составляет тысячи единиц и продолжает быстро увеличиваться. Зондовый микроскоп был изобретен в начале 80-х годов прошлого столетия [1]. Доступные для широкого круга исследователей приборы стали выпускаться только в 90-х годах [2–4]. Таким образом, метод зондовой микроскопии крайне молод. Вследствие этого имеются очевидные трудности в интерпретации получаемых результатов измерений рельефа объекта с нанометровым разрешением [5].

Основным способом визуализации измерений в зондовой микроскопии является двумерное графическое изображение оцифрованных значений высот, получаемого в процессе построчного сканирования изучаемого объекта зондом микроскопа.

Для снижения уровня шумов, улучшения визуального восприятия, увеличения видимого разрешения, в специализированных программных комплексах визуализации измерений зондовой микроскопии к полученному изображению применяют фильтры, сходные с аналогичными из известных графических пакетов [6]. В связи с тем, что такие средства обработки изображений не учитывают особенности процесса функционирования микроскопа, при их применении следует принимать во внимание возможность появления артефактов [7]. В случае туннельной микроскопии выявление артефактов связано с отсутствием априорного знания геометрии иглы и сканируемой поверхности объектов ввиду экстремальной малости нанообразований.

Одним из факторов, влияющих на точность измерений, является размеры и форма иглы зондового микроскопа [8]. Учет геометрии острия при интерпретации результатов сканирования делает необходимым развитие программных методов с более углубленной обработкой данных, опирающейся на сложные физические модели процесса зондовой микроскопии.

Возникающие при этом задачи не могут быть сведены к чисто вычислительным, так как стандартные методы решения возникающих уравнений требуют неприемлемо больших вычислительных ресурсов. Использование неявных методов и приводит к новым моделям программного обеспечения, которые обеспечивают обработку комплексных данных, включающих не только измерения, но и параметры процесса, при котором они были получены.

В зависимости от условий, которые учитываются при обработке результатов сканирования, процесс измерений может описываться разными физическими моделями. Это делает необходимым реализацию такой архитектуры программного комплекса, которая допускает замену модуля вычислений

физической модели. В более общем случае эта задача была решена ранее в системе САФРА [9], которая обеспечивала сборку пакета программ применительно к задачам интерпретации результатов измерений физического эксперимента.

Другое решение реализуется средствами открытой архитектуры [10]. Они предусматривают использование объектно-ориентированного подхода к проектированию программы и описание спецификаций программных модулей. Реализация принципов открытой архитектуры в полном объеме требует достаточно высокой квалификации разработчика и является достаточно трудоемкой задачей. В случае программных продуктов, используемых для исследовательских целей, в которых предполагается заменять только модуль расчетов по физической модели, эти трудозатраты вряд ли оправданы.

В диссертации предлагается решение этой задачи стандартными возможностями языка C# в среде .NET. Возможности языка обеспечивают достаточно простые решения большинства вопросов, связанных с реализацией замены модуля в программном комплексе большого объема: автоматическое освобождение памяти и учет специфических особенностей различных версий операционной системы WINDOWS и др.

Современные средства вычислительной техники характеризуются высокой вычислительной мощностью, которая обеспечивается за счет многоядерности процессоров. Использование многоядерности достигается средствами многопоточной обработки информации. Решаемая в диссертации задача учета геометрии острия иглы делает возможным эффективное распараллеливание вычислений за счет построчной обработки результатов сканирования. Примененные в работе решения реализованы стандартными средствами языка C#.

Диссертация включает Введение, 4 главы и Заключение. Список цитируемой литературы содержит 50 позиций.

В первой главе приведен обзор наиболее распространенных программных продуктов для интерпретации измерений рельефа поверхности зондово-

го микроскопа. Выполненное сравнение позволяет сделать вывод о том, что их функциональные возможности в целом совпадают. Рассмотренные программные комплексы содержат модули первичной обработки измерений, которые позволяют уменьшить уровень шумов, модули двухмерной и трехмерной визуализации измерений, модули преобразования изображений, которые улучшают видимость деталей рельефа, модули коррекции геометрических искажений, вызванных особенностями конструкции пьезосканера, и модули структурного и количественного анализа формы рельефа.

Основным выводом этой главы является необходимость дополнения программных средств интерпретации измерений алгоритмами учёта влияния формы острия иглы на изображение рельефа, получаемого с помощью туннельного и атомно-силового микроскопов высокого разрешения.

Разработанный алгоритм решения этой задачи описывается во второй главе. Предполагается, что известны измерения рельефа поверхности подложки и параметры сканирования, которые позволяют с помощью существующих моделей туннельного тока вычислить величину туннельного зазора. Если предположить, что геометрия острия иглы известна, то рельеф сканируемой поверхности определяется из геометрических вычислений. Отличие разработанного алгоритма от известных состоит в том, что форма и размеры острия зонда и рельеф поверхности определяются в нем одновременно.

В третьей главе диссертации дается обоснование решений, использованных при построении программного комплекса для интерпретации данных сканирования рельефа образца с учетом геометрии острия иглы туннельного микроскопа.

В четвертой главе диссертации описываются программная реализация комплекса и средства пользователя для настройки процесса обработки данных. Приводятся результаты применения программного комплекса в исследованиях, проводившихся в Российском химико-технологическом университете им. Д.И. Менделеева.

Глава 1. ОБЗОР ПРОГРАММНЫХ СРЕДСТВ ОБРАБОТКИ РЕЗУЛЬТАТОВ СКАНИРОВАНИЯ ПОВЕРХНОСТИ ТУННЕЛЬНЫМ МИКРОСКОПОМ

В главе дается обзор наиболее распространенных программных продуктов для интерпретации измерений рельефа поверхности, выполненных с помощью зондового микроскопа. Большинство из них поставляются в составе зондового микроскопа фирмой-производителем. Поэтому неудивительно, что большинство функций этих программных продуктов совпадают. Существенные отличия имеют место только в составе средств, предназначенных для анализа особенностей изображений [11].

Средства работы с изображениями в целом не выходят за рамки традиционных настраиваемых графических фильтров. Осуществляемые преобразования не используют особенности физического процесса сканирования. Следует заметить, что хотя такие преобразования позволяют существенно увеличить видимое разрешение изображения и улучшить различимость объектов на поверхности, тем не менее, они могут служить причиной появления артефактов.

§1. Зондовая микроскопия рельефа поверхности

Зондовые микроскопы для измерения молекулярного или даже атомного рельефа поверхности [1] включают два класса приборов: атомно-силовой и туннельный микроскопы. В обоих приборах молекулярно-атомный рельеф поверхности исследуемого образца измеряется с помощью зонда – иглы острие, которой имеет радиус закругления порядка нанометров. При приближении зонда к поверхности образца возникает сигнал взаимодействия молекул материала иглы зонда с молекулами образца и измеряются три координаты его позиции в системе координат микроскопа.

Движение иглы осуществляется с помощью пьезодвигателей с пространственным разрешением более чем на порядок лучше значения радиуса закругления иглы. Движение зонда по некоторой траектории позволяет получить определенное представление о рельефе, образованном молекулами на поверхности образца, вдоль этой траектории. Чем тоньше острие зонда и точнее движение его по траектории, тем с более высоким пространственным разрешением измеряется рельеф поверхности.

В зондовых микроскопах сканирование поверхности обычно осуществляется с помощью линейной развёртки - строка за строкой. Шаг строк и частота измерения высоты рельефа формируются системой управления микроскопа по заданному разрешению и размеру поля сканирования в пределах диапазонов управления движением пьезодвигателей.

В атомно-силовом микроскопе игла находится на конце тонкой упругой балочки, называемой *кантилевером*. При сканировании кантилевер вводят в режим автоколебаний с частотой до сотен килогерц. Амплитуда колебаний зависит от сил, действующих на кантилевер, и достигает порядка 100 нм. Оптико-механическая система измеряет частоту линейных и крутильных колебаний кантилевера, которые затем пересчитываются в линейные и угловые перемещения иглы относительно основания кантилевера. Система управления перемещает кантилевер по вертикали, чтобы амплитуда вынуж-

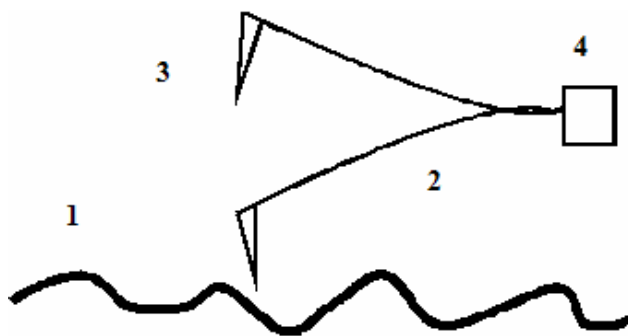


Рис. 1.1. Принцип измерения рельефа атомно-силовым микроскопом:
1 – исследуемая поверхность, 2 - колеблющийся кантилевер, 3 - игла,
4 - возбудитель колебаний.

денных колебаний оставалась постоянной. В системе координат пьезодвигателей положение основания кантилевера неизменно, поэтому относительную координату высоты рельефа можно вычислить, если известно смещение пьезодвигателя и вертикальная и боковая амплитуды колебаний иглы (рис. 1.1).

В туннельной микроскопии система управления поддерживает острие иглы над исследуемой поверхностью на расстоянии туннельного зазора (рис. 1.2). Под туннельным зазором понимается расстояние от иглы до молекул поверхности образца, на котором существует туннельный ток заданной величины при фиксированном напряжении. При импульсных изменениях напряжения величина туннельного зазора может варьироваться [12], поэтому туннельный микроскоп обычно работает при постоянном напряжении. Величина туннельного зазора вычисляется с помощью физических моделей по напряжению и туннельному току, которые программно задаются при сканировании.

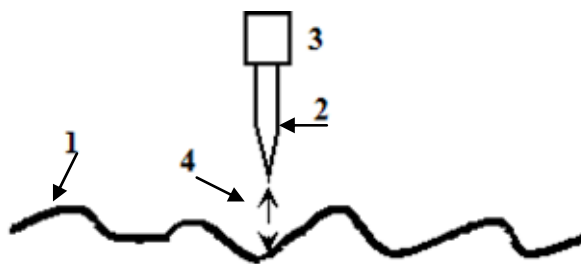


Рис. 1.2. Принцип измерения рельефа туннельным микроскопом:

1 – исследуемая поверхность, 2 – игла, 3 – пьезопровод,
4 – туннельный зазор.

Результаты измерений обычно представляют в виде цветного изображения. Кодирование высоты осуществляется в условных цветах с помощью выбранной создателем программного обеспечения палитры. Например, в рассмотренном ниже программном пакете StmShell низкие участки окрашены в синие цвета, высокие – в красные. Такая раскраска используется независимо от амплитуды перепада высот измеренного рельефа. В результате для по-

верхностей с разными перепадами высот набор цветов на изображениях, полученных в результате сканирования, оказывается одинаковым.

Заметим, что на изображениях с подобным принципом раскраски человеческий глаз не различает переходные цвета. Изображение воспринимается таким, как будто оно раскрашено лишь в 4 основных цвета (синий, зеленый, желтый и красный), которые имеют яркостные оттенки, причем только границы между этими цветами хорошо различимы. Эти границы фактически определяют линии уровня высоты. Для улучшения восприятия, более совершенным методом изображения высоты рельефа условными цветами следует считать раздельное использование одноцветной палитры и механизма управления отображением линий уровня. Подобный способ раскраски используется в программном комплексе StmShell и в разработанном в рамках данной работы макете программных средств туннельного микроскопа.

§2. Программные пакеты для отображения измерений рельефа поверхности

Программное обеспечение для сканирующих зондовых микроскопов (СЗМ) в настоящее время в достаточной степени устоялось. Большинство пакетов программ для работы с СЗМ предоставляет примерно одинаковые возможности. При этом, поскольку для увеличения быстродействия нижний уровень управления СЗМ обычно «зашит» в чип микроконтроллера, на программное обеспечение верхнего уровня в настоящее время ложатся только три задачи:

1. Настройка контроллера, отвечающего за процесс сканирования. Для шагового контроллера пьезодвигателей возможно указать размеры одного шага перемещения, скорость перемещения и направление движения.
2. Низкоуровневая обработка сигнала, поступающего в ходе измерений. Для обработки компонентов сигнала обычно применяется медианный фильтр, который состоит в отбрасывании максимумов и минимумов

сигнала и усреднении оставшейся части измерений [13].

3. Обеспечение интерфейса с оператором микроскопа для задания параметров сканирования и выбора метода наглядного представления получаемых результатов. При этом решаются следующие задачи:

1. Улучшение субъективного восприятия изображения поверхности путем удаления шумов и подчеркивания деталей.
2. Выделение на изображении поверхности объектов, представляющих интерес.
3. Измерение различных геометрических характеристик изображения объектов: длины, ширины, диаметра, периметра и площади.

Программное обеспечение туннельных микроскопов включает средства обработки изображений, полученных в результате измерений рельефа поверхности исследуемого образца [14]. Большая часть из них совпадает со стандартными средствами работы с графическим изображением [15]. Во все пакеты включены примерно совпадающие наборы программных средств обработки изображений, которые учитывают специфику технологии получения изображения в зондовой микроскопии.

Рассмотрим более подробно функции обработки сигнала в пакетах программного обеспечения для работы со сканирующей зондовой микроскопией. Для анализа были выбраны две отечественные разработки СЗМ: NT-MDT Nanoeducator и FemtoScan Online и одна зарубежная – Nanotechnology StmShell, используемая в отечественном сканирующем туннельном микроскопе (СТМ) НПО «АЛМАЗ». Функции и интерфейс настройки и управления этими СЗМ не анализируются. Они приведены ниже для иллюстрации действий, выполнение которых необходимо для получения сигнала сканирования нано- или микрорельефа поверхности исследуемого образца и обстоятельств, влияющих на содержание такого сигнала.

2.1. NT-MDT Nanoeducator

Зондовый микроскоп Nanoeducator разработан отечественной фирмой NT-MDT для образовательных целей [3; 16]. Он может работать в двух ре-

жимах: как зондовый, так и туннельный микроскоп. Возможности программного обеспечения прибора ограничены функциями двух модулей – «модуля управления» и «модуля обработки изображения».

Основные функции модуля управления

1. Процедура подготовки к измерениям (предварительный подвод зондового датчика к образцу, построение резонансной кривой и установка рабочей частоты колебаний пьезодатчика с укрепленным на нем зондом).
2. Задание и просмотр текущих параметров (рис. 1.3), установленных в конфигурации измерительной головки и контроллера.
3. Задание и просмотр текущих параметров планируемых измерений и процедур.
4. Включение видеомикроскопа, осмотр исследуемого образца и контроль взаимного расположения зонда и образца (управление расстоянием между образцом и кончиком зонда сканера).
5. Управление процедурой подвода сканера к образцу.
6. Управление процедурами сканирования и спектроскопических измерений (измерение отклика поверхности образца в ответ на приложенную силу давления со стороны зонда атомно-силового микроскопа)
7. Автоматизированные отчеты («Report generator»). Данные отчеты содержат изображения, обработанные пользователем после сканирования, параметры сканирования, список параметров примененных фильтров и комментарии пользователя.

Оставляя в стороне вопрос разумности англоязычного интерфейса в отечественной системе обучения, заметим, что функция подавления амплитуды колебаний иглы в правом окне (оно вызывается при нажатии кнопки «Set interaction» левого окна) должна влиять на получаемое значение высоты элементов исследуемого рельефа. Хотя это влияние, похоже, может быть ослаблено установкой параметра глубины обратной связи (функцией «Feed back loop gain» левого окна) метрологические характеристики при этом остаются неясными, и сигнал данного СЗМ даёт, скорее, качественное, а не количест-

венное знание параметров рельефа подложки.

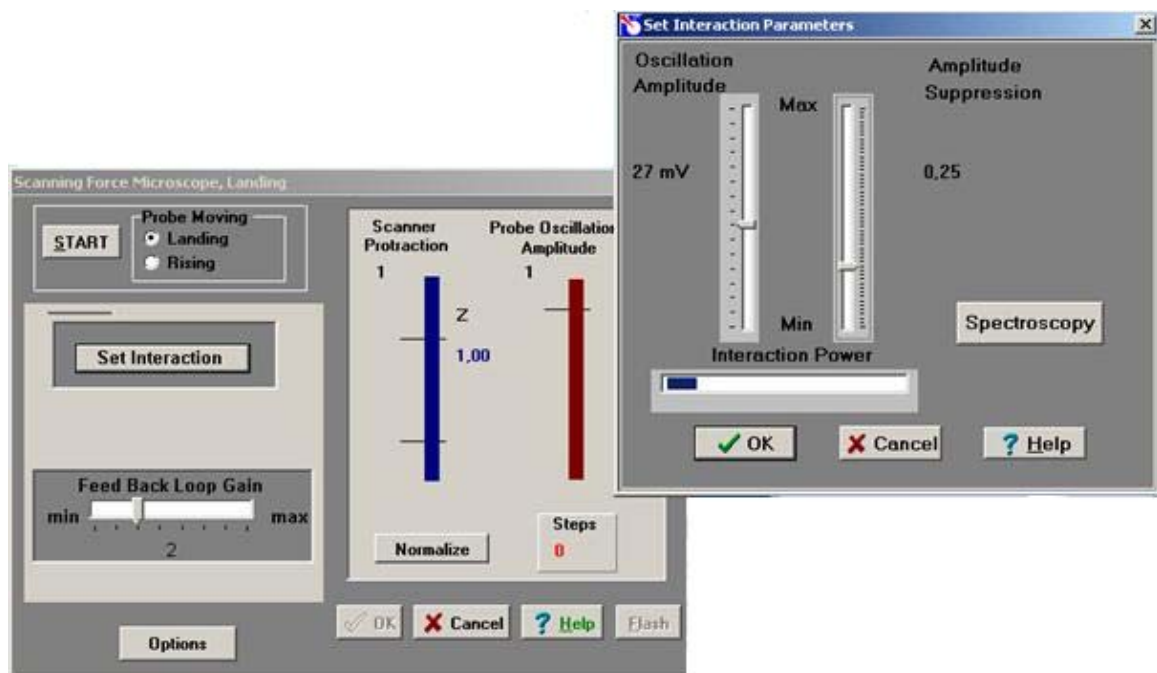


Рис. 1.3. Снимки окон модуля управления Nanoeducator.

Модуль обработки изображений позволяет:

1. Представлять данные в двухмерном и трехмерном виде различными вариантами искусственной подсветки (рис. 1.4).
2. Проводить статистическую обработку данных (гистограмма яркости изображения, вычисление величины средней и среднеквадратичной шероховатости полученного рельефа поверхности [17]).
3. Использовать несколько видов фильтрации, включая градиентную, сглаживающую и Фурье-фильтрацию.
4. Приводить изображение к плоскому виду и строить сечения.

В приведенном рисунке выполняется простейшее исследование образца размером 5мкм×5мкм, полученного построчечным зондовым сканированием с шагом 50 нм. Скорость прямого хода 320 шаг/сек, обратного – 640 шаг/сек. Таким образом, время 10000 шагов сканирования данного образца занимает три четверти минуты, и с учётом установки 10 параметров управления (см. рис. 1.3) полное время получения сигнала составляет не менее двух минут.

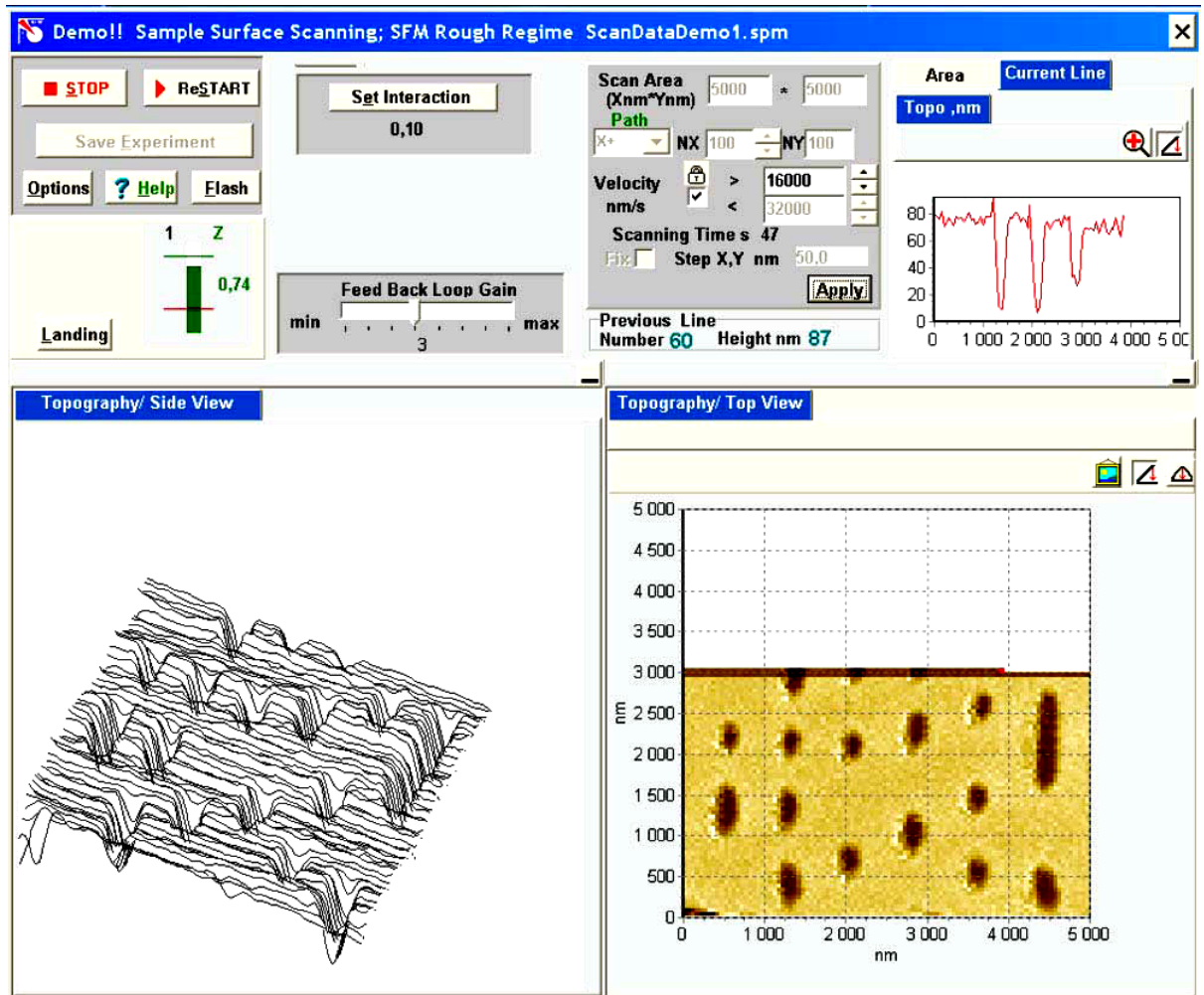


Рис. 1.4. Вид окна модуля обработки изображений Nanoeducator.

Наблюдаемые этим прибором неровности (провалы) рельефа имеют вдоль строки размер не меньше четверти микрона. Они все вытянуты в междустрочном направлении до размеров $0,5 \div 1,5$ мкм. Их глубина при этом достигает 90 нм с шириной дна вдоль строки порядка 1 шага (50 нм) при наблюдаемой амплитуде шума сигнала высоты порядка 10 нм.

2.2. FemtoScan Online

Отечественный прибор ФемтоСкан сконструирован НПП "Центр перспективных технологий" в 1990-х годах преимущественно для биомедицинских применений [18]. Микроскоп разрабатывался для решения задач длительного или постоянного наблюдения за процессами в микробиосистемах и поэтому обладает многопользовательским режимом и возможностью полного управления по сети.

Также как и NanoEducator, ФемтоСкан позволяет осуществлять сканирование в режиме атомно-силового и туннельного микроскопа. Возможности ФемтоСкан Онлайн,- программных средств микроскопа ФемтоСкан,- сравнимы со средствами прибора NanoEducator.

Возможности модуля управления. Отличительная особенность модуля управления ФемтоСкан Онлайн от NanoEducator заключается в том, что управление реализовано в виде сервера (рис. 1.5), к которому подключаются клиентские компьютеры, и с которых операторы формируют задания на сканирование. При этом каждый оператор самостоятельно задает свои параметры конфигурации и калибровки (аналогичные параметрам Nanoeducator), а сервер (т.е. управляющий модуль) может лишь автоматически корректировать некоторые параметры. Из протокола работы на рис. 1.5 видно, что потоки управления прибором так же, как и в предыдущем примере, имеют средний интервал около 30 сек.

Рис. 1.6 иллюстрирует некоторые параметры процесса СТМ. В туннельном зазоре напряжение 9 В, туннельный ток 43 пА ($2,684 \cdot 10^8$ электронов/сек), размер сканируемого участка поверхности - 2,5 мкм, число строк 512 и число точек в строке - 512 (шаг равен 5 нм), скорость сканирования 7,3 строк/сек. При этих параметрах измерения полное время сканирования составляет 70 сек.

Клиент-оператор может пользоваться следующими возможностями управления работой микроскопа:

1. Быстрый подвод и отвод столика к зонду/от зонда.
2. Включение/выключение питания двигателя.
3. Загрузка программы сигнального процессора.
4. Вывод сигнала, приходящего от микроскопа ("осциллограммы").

Возможности модулей обработки измерений и анализа.

Программное обеспечение *ФемтоСкан Онлайн* состоит из одного главного (основного) модуля и нескольких дополнительных модулей [2].

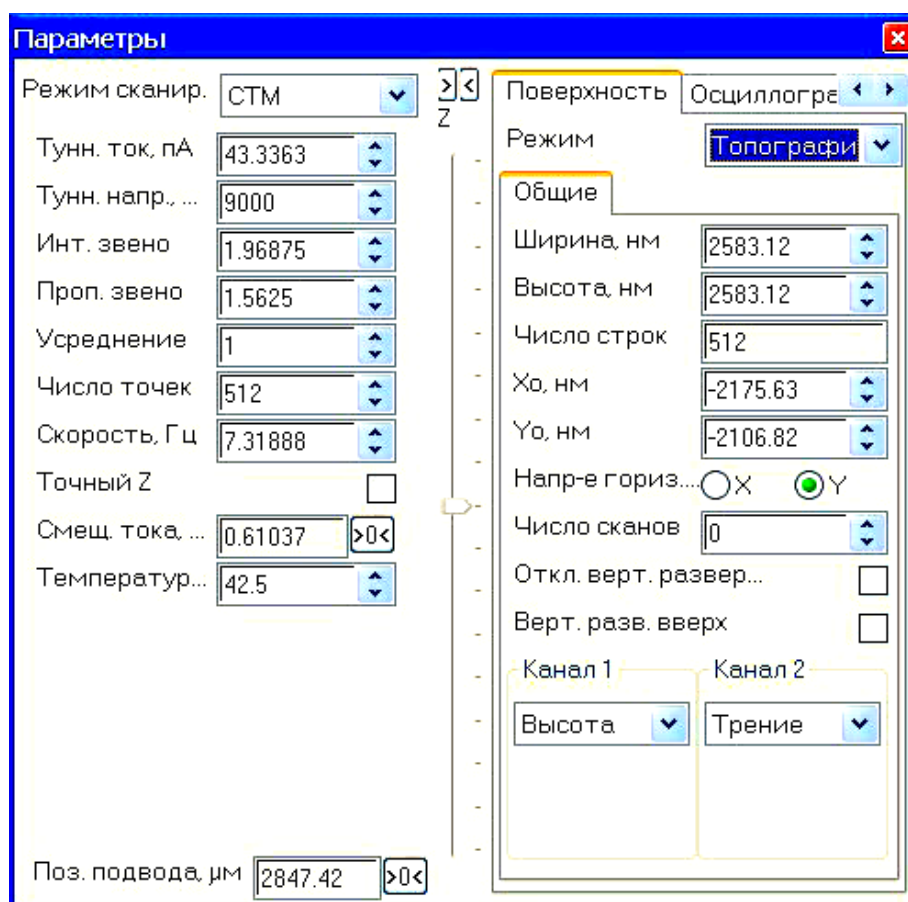


Рис. 1.6. Настройки программы FemтоСкан Онлайн на компьютере оператора.

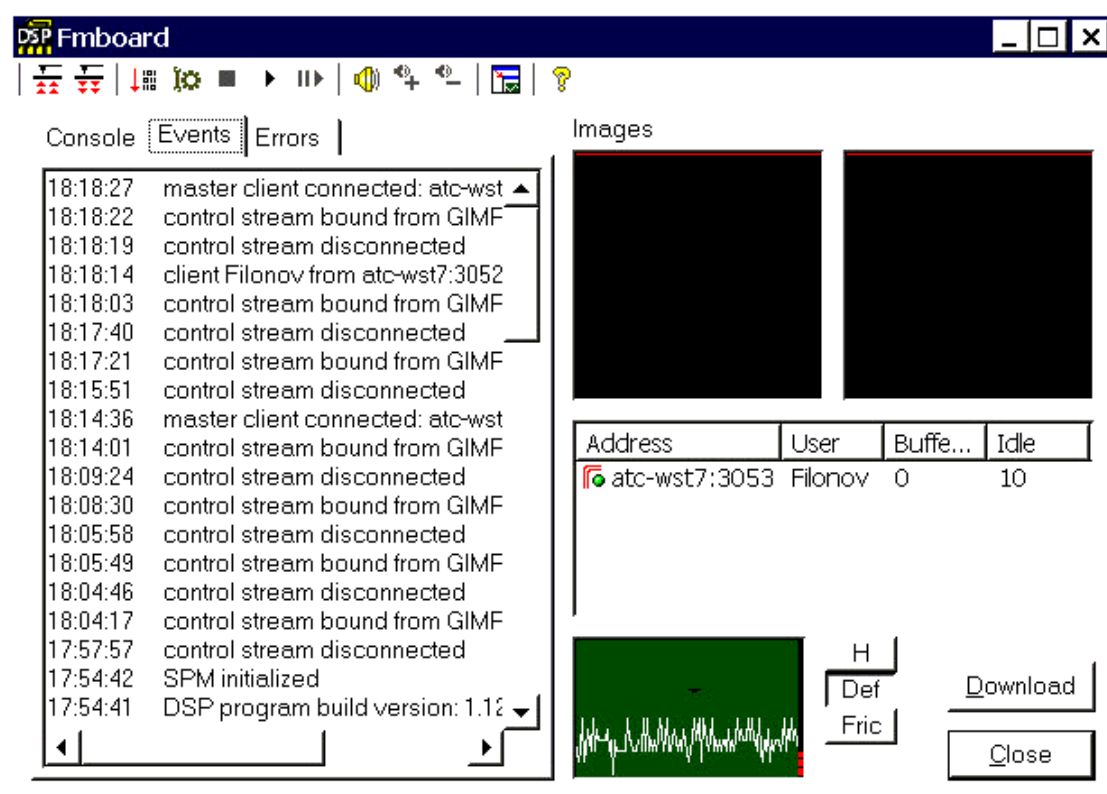


Рис. 1.5. Серверный интерфейс FemтоСкан Онлайн.

Основной модуль ФемтоСкан Онлайн обеспечивает богатый набор следующих действий преобразований изображений осциллограммы:

1. Представление данных в двухмерном и трехмерном виде с различными вариантами искусственной подсветки (рис. 1.7).
2. Простейшая обработка изображений.
 1. Обрезание изображения до требуемых размеров.
 2. Инвертирование изображения.
 3. Увеличение резкости в выбранной области маской шириной 3 пиксела.
 4. Транспонирование изображения относительно оси X или Y.
 5. Поворачивание изображения на произвольный угол.
3. Фильтрация измерений.
 1. Усреднение выбранной области с определенной шириной.
 2. Осуществление пороговой фильтрации измерений.
 3. Применение фильтра Фурье к измерениям рельефа.
 4. Применение фильтра Винера с маской произвольного размера.
 5. Применение медианного фильтра:
 1. Задание линий/столбцов из измерений рельефа поверхности для выполнения медианной фильтрации размером 3x3 пиксела.
 2. Выполнение медианной фильтрации выбранной области размером 3x3 пиксела.
4. Удаление среднего наклона поверхности образца, оцененного методом наименьших квадратов.
5. Удаление крупного рельефа вычитанием кубического В-сплайна осреднённой осциллограммы.
6. Сглаживание участка с сохранением структуры поверхности.
7. Статистический анализ изображения.
8. Преобразование изображений:
 1. Перемещение, изменение размеров, уменьшение изображений.
 2. Масштабирование по оси Z.
 3. Изменение горизонтальных масштабов.

4. Копирование изображения или его части.
5. Автоматическая или ручная картография цветов.
9. Измерение различных параметров изображения:
 1. Шероховатости.
 2. Размеров квадратичной области на поверхности.
 3. Расстояния между указанными объектами.
 4. Длины кривой.
 5. Длины изолинии, площади и объема ограниченной ею области.
 6. Расстояния, угла, высоты, суммарного квадрата в областях.
 7. Части квадрата, соответствующего выбранным высотам в гистограммах.
10. Измерение параметров иглы путем калибровки с использованием тестовой поверхности.

Возможности дополнительных модулей.

1. Модуль обработки сигналов СЗМ.
 1. Усреднение по строкам (приведение линии просматриваемого изображения к соответствующему среднему уровню, вычисляемому по всему набору измерений, либо по произвольной области).
 2. Исправление искажений измерений, вызванных гистерезисом пьезопривода.
 3. Нормализация силовой кривой (приведение кривой силы к нормализованному виду) для АСМ.
 4. Построение кривой разделения величины силы.
 5. WLC (Worm-like chain) анализ: нахождение локальных минимумов на силовой кривой.
2. Обработывающий модуль просвечивающей электронной микроскопии (ПЭМ).
 1. Выделение зерен.
 2. Выделение границ методом Канни.
 3. Градиентное преобразование.

4. Линейный фильтр Лапласа с матрицей 3x3 пиксела.
 5. Линейный фильтр Гаусса с матрицей 3x3 пиксела.
3. Модуль, обрабатывающий информацию в соответствии со стандартами ГОСТ: стандарты классов шероховатости (ГОСТ 2789-59), значений параметра шероховатости (ГОСТ 2789-73, 9378-93) и стандарт средств измерений шероховатости поверхности профильным методом (ГОСТ 19300-86).
1. Анализ шероховатости поверхности рельефа (вычисление параметров шероховатости с помощью поперечных сечений поверхности).
 2. Анализ шероховатости кривой.
 3. Высота по интерференционной картине (расчет крутизны наклона «холмика» на поверхности кристалла, получаемого из интерференционной картины сигнала).

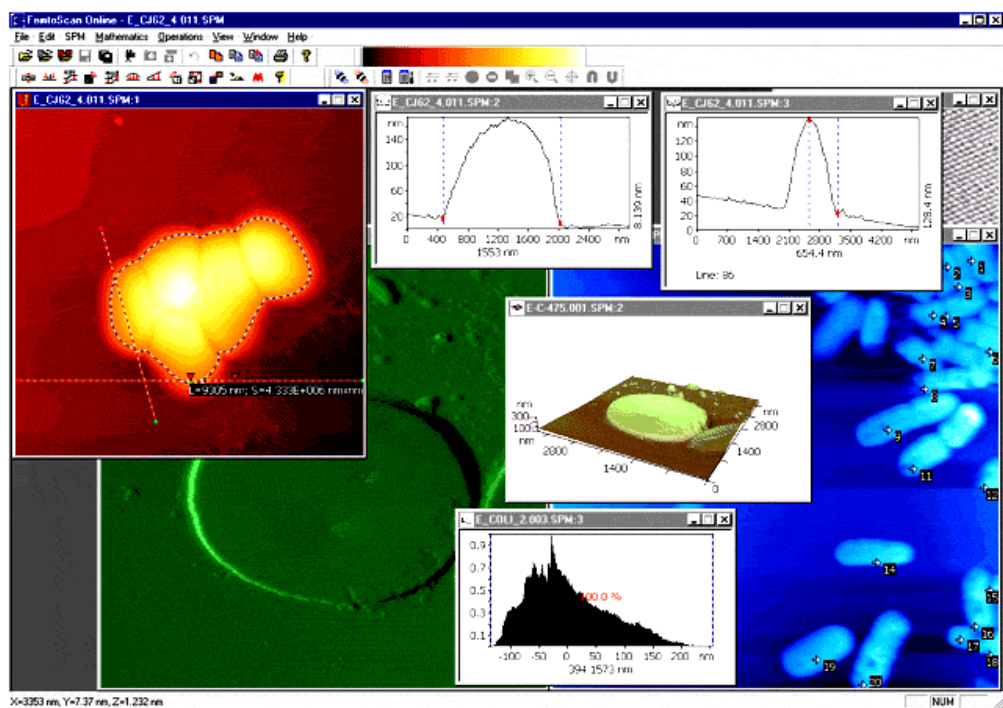


Рис. 1.7. Снимок экрана программы ФемтоСкан.

На рис. 1.7 показан результат обработки программой ФемтоСкан изображения сигнала, приведенного в левом верхнем окне экрана монитора. В этом окне видны два сечения изображения, им соответствуют два верхних графика. Кроме того, показано трехмерное изображение рельефа объекта, видимого в левом нижнем окне, окно с распределением точек изображения

рельефа образца по высотам, изображение регулярного рельефа поверхности кристалла и микроскопическое изображение палочковидных бактерий с нанесенными на них порядковыми номерами.

Хорошо видно, что интерфейс пользователя включает богатый набор около 40 кнопок управления, которые помимо десятка стандартных функций работы с файлами и изображениями содержат вызовы функций управления процессом сканирования, обращение к функциям математических преобразований и несколько десятков специальных операций работы с изображением сигнала.

2.3. Программное обеспечение СТМ производства НПО «АЛМАЗ»

Для туннельного микроскопа производства НПО «АЛМАЗ» используется программный продукт Nanotechnology StmShell. Пакет программ StmShell также, как ФемтоСкан и NanoEducator представляет собой интегрированное решение для проведения процесса сканирования, литографии и обработки результатов сканирования.

Упомянутая здесь атомная наноперьевая нанолитография – это способ построения произвольных молекулярных структур на поверхности подложки иглы микроскопа. Сканирующий зонд передвигается над поверхностью, оставляя за собой линии атомных размеров. В атомной литографии внутренние и внешние степени свободы индивидуальных атомов контролируются с нанометровой точностью внешними электромагнитными полями и требуемый нанообъект «собирается» из индивидуальных атомов, молекул, биологических клеток и т.п. [19].

Остановимся подробнее на важном для целей данного исследования модуле обработки и анализа изображений пакета программ StmShell. Возможности этого модуля следующие:

1. Представление данных в двухмерном виде (вид рельефа поверхности сверху) с использованием различных палитр условных цветов высот.
2. Фильтрация измерений.

1. Медианный фильтр.
2. Пороговый фильтр с точной ручной настройкой верхнего и нижнего уровней.
3. Устранение среднего наклона поверхности.
 1. Просмотр изображений.
 2. Изменение размера изображения.
 3. Приведение изображения к масштабу «1 к 1».
 4. Переключение палитры цветов для отображения изображения рельефа поверхности.
4. Сохранение исходных («сырых») данных сканирования в файле.

Ниже на рис. 1.8 показано рабочее окно программы с открытым документом, представляющим собой изображение меди, напыленной на стеклянную подложку (слева) и результат его обработки (справа). Получение такого сигнала сканированием в данном случае 256 вертикальных строк по 256 точек в поле зрения 0,5 мкм×0,5 мкм требует около 12 мин. В каждой точке измерения проводятся 4 раза. Накопленные измерения используются в про-

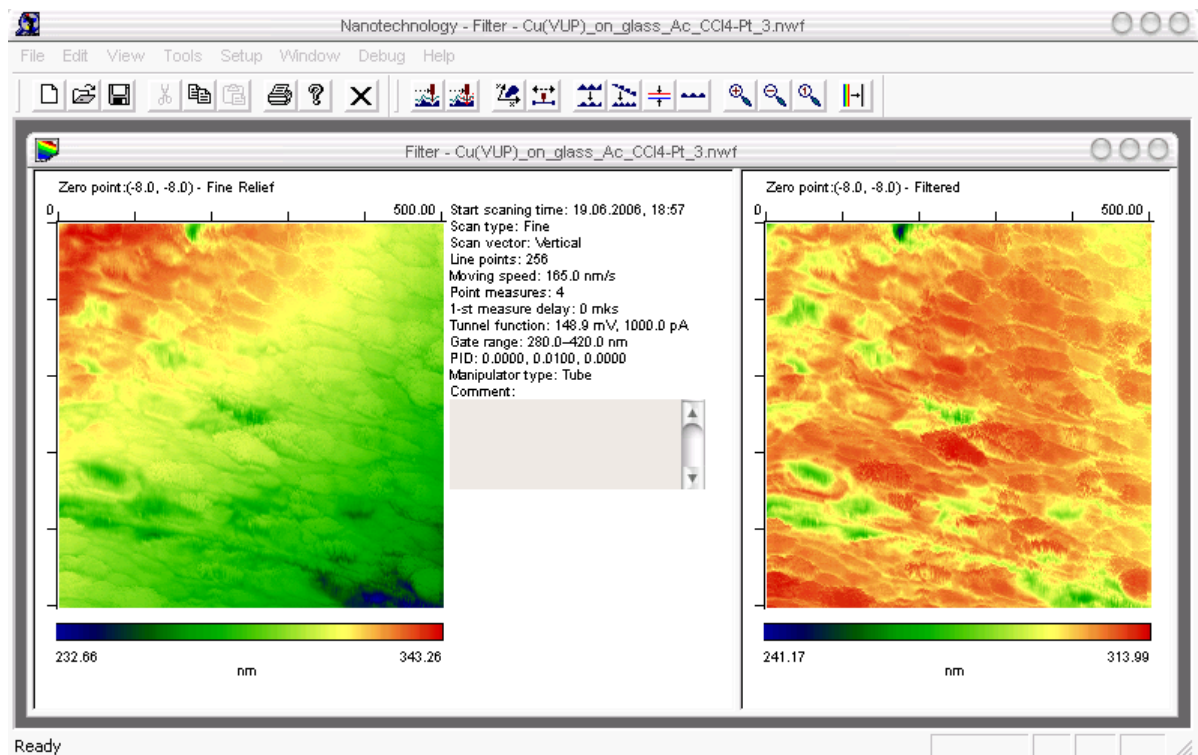


Рис. 1.8. Окно программы StmShell.

граммном фильтре входного сигнала. Рядом указаны параметры туннельного тока (1 нА, 149 мВ) и рабочий диапазон высот 140 нм.

Следует обратить внимание на улучшение детализации изображения после учёта наклона поверхности, дающего перепад высот (38 нм).

Общим свойством программных моделей описанных комплексов является возможность вести обработку данных в той последовательности, которая удобна пользователю. Это оказалось возможным благодаря тому, что каждая команда реализует применение к имеющимся данным функции из фиксированного набора, и тем, что результат может быть оценен визуально. Участие пользователя в настройке преобразования состоит в задании его параметров, которое выполняется с помощью оконных органов управления. Таким образом, если данные, которые получаются после выполнения i -той команды обозначить через R_i , функцию преобразования данных – через F_i , то для всех i можно рекурсивно записать:

$$R_i = F_i(R_{i-1}, P_i),$$

где P_i – множество параметров преобразования данных на шаге i .

Итак, описанные программные средства зондовой микроскопии отличаются тем, что в них реализована попытка извлечь искомый рельеф исследуемой поверхности путём фильтрации и преобразования исходного изображения, *без существенного анализа обстоятельств его получения*. Ниже анализируются применяемые на практике способы улучшения качества построения знания о нанорельефе поверхности исследуемого образца алгоритмическим учётом обстоятельств получения исходного сигнала.

§3. Увеличение визуального разрешения нанорельефа

3.1. Связь наклона рельефа с разрешением его глубины

Участки поверхности, которые исследуются с помощью зондовых микроскопов, имеют достаточно малые размеры: от сотен нанометров до не-

скольких микрон. Это является одной из причин того, что достаточно часто исследуемый участок поверхности имеет некоторый наклон.

При наклоне образца пропадает наблюдаемость глубин рельефных неровностей. Кроме этого следует иметь в виду, что весь диапазон возможных высот на изображении часто кодируются весьма малым диапазоном видимых цветов. Поэтому даже при незначительном наклоне рельефа мелкие образования на поверхности образца оказываются плохо различимыми в их цветовом представлении.

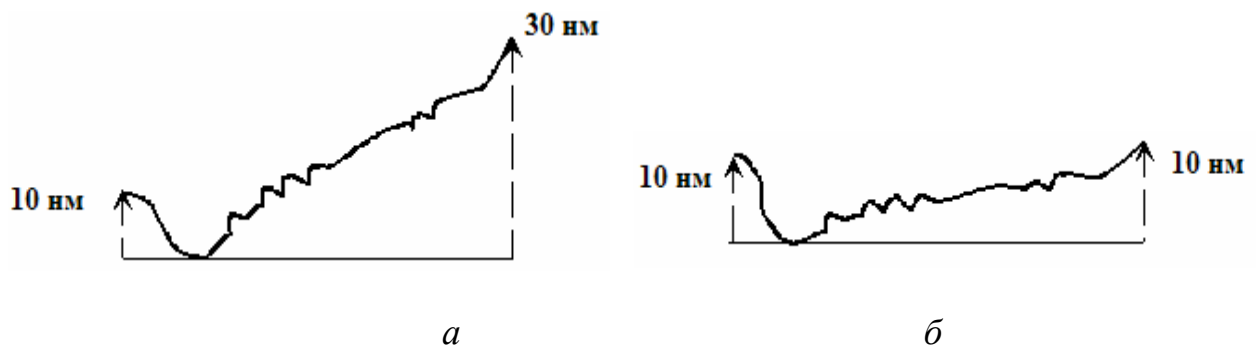


Рис. 1.9. Уменьшение отношения перепада высот к размерам нанобъектов.
а - исходное изображение, *б* – результат вычитания общего наклона.

Рассмотрим в качестве примера срез рельефа поверхности, изображенный на рис. 1.9 *а*. Перепад высот на изображенном участке достигает 30 нм. Если образование имеет высоту 3 нм, что составляет 1/10 перепада высот исследуемого участка, то, в силу отмеченной выше особенности восприятия глазом используемой раскраски, образование и соседние с ним участки поверхности окажутся целиком окрашены одним цветом, что сделает образование трудно различимым на фоне поверхности.

После вычитания из измерений рельефа поверхности общего наклона, перепад высот уменьшится в 3 раза. Размер нанообразования составит 1/3 перепада высот. Принимая во внимание, что в изображениях поверхности, получаемых с помощью StmShell, хорошо различимы только 4 цвета, основание и вершина нанообразования окажутся окрашенными в разные цвета, и оно станет легко заметным.

3.2. Методы удаления среднего наклона

В работе [1] предложены несколько способов учета общего наклона поверхности. На первом, общем для всех способов этапе методом наименьших квадратов находится аппроксимирующая плоскость $P^{(1)}(x,y)$, имеющая минимальные отклонения от рельефа поверхности $Z = f(x,y)$ (рис. 1.10). Затем производится вычитание данной плоскости из СЗМ изображения. Эту операцию можно выполнить различными способами.

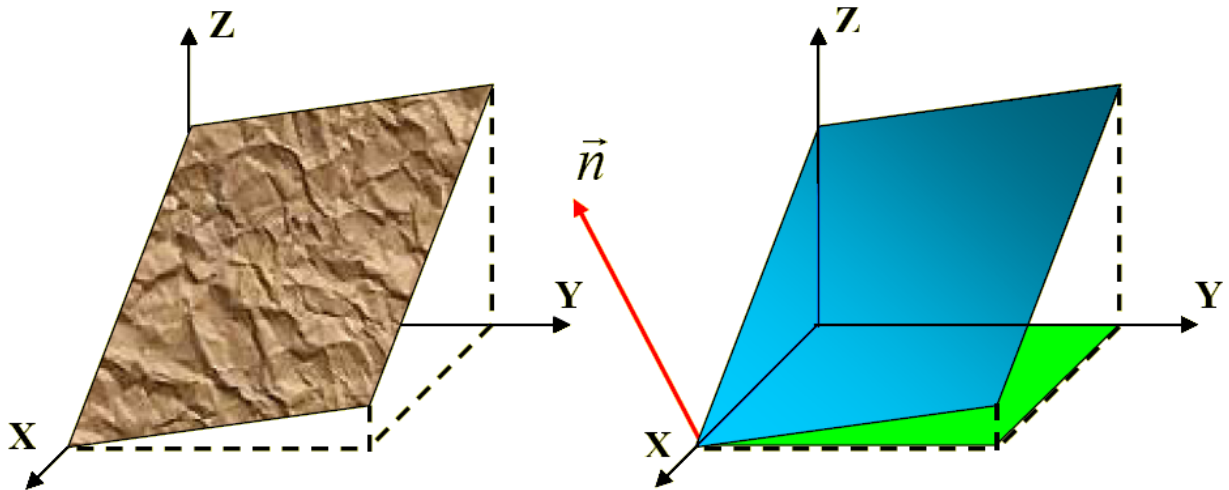


Рис. 1.10. Вычитание постоянного наклона из СЗМ-изображения поверхности.

Первый способ состоит в вычитании из рельефа поверхности плоскости вдоль нормали n . Получившаяся поверхность будет выглядеть так, если бы мы смотрели на исследуемую поверхность вдоль вектора нормали. Недостатком этого способа является искажение геометрических размеров нанобразований, связанное с кривизной острия иглы.

Второй способ состоит в вычитании z - координат плоскости из z - координат СЗМ-изображения:

$$Z'_{ij} = Z_{ij} - P_{ij}^{(1)}.$$

Этот прием приводит к изменению геометрических соотношений в плоскости X,Y между объектами в СЗМ-изображении. Смысл такого преобразования (в котором информация о геометрии всего образца отсутствует) состоит в том, что по полученному изображению можно судить об относительных размерах нанобразований, поскольку в результате получается поверх-

ность с меньшим (в несколько раз) перепадом высот, что, как было сказано, делает мелкие детали изображения, более заметными (рис. 1.11).

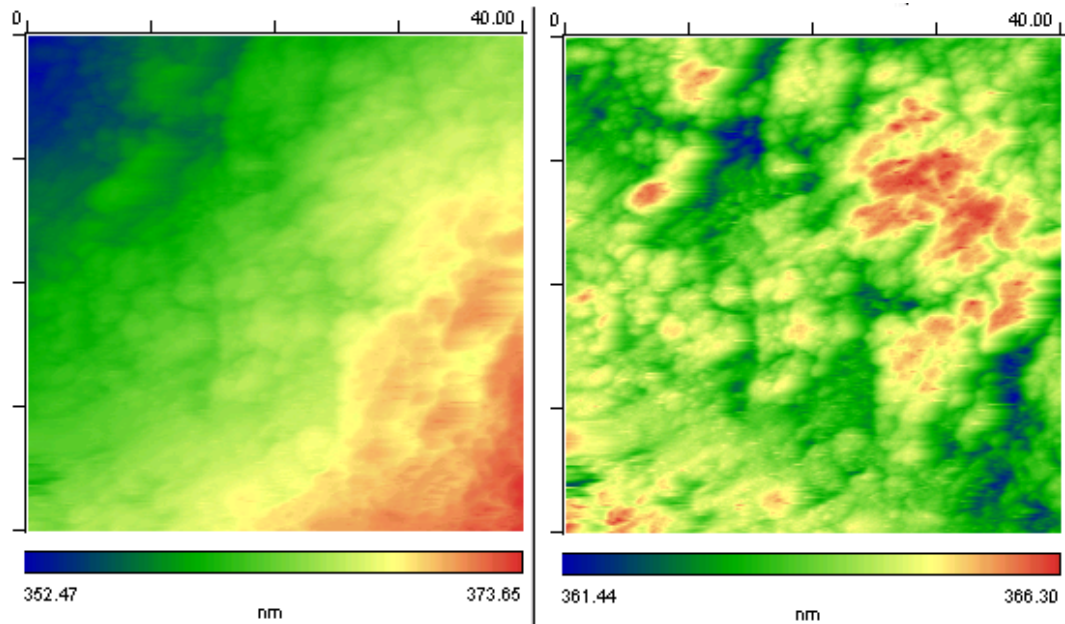


Рис. 1.11. Удаление среднего наклона.

а - исходное изображение, б - изображение с удаленным наклоном.

§4. Искажения, связанные с неопределённостью модели сканера

Неидеальность свойств пьезосканера СЗМ приводит к тому, что получаемое изображение содержит ряд специфических искажений [20]. Неидеальности сканера, такие как неравноправность прямого и обратного хода сканера (гистерезис), крип (см. ниже) и нелинейность пьезокерамики, частично компенсируются аппаратными средствами и выбором оптимальных режимов сканирования [21]. Однако некоторые из погрешностей измерений нельзя учесть в процессе сканирования. В рассматриваемых комплексах некоторые из таких погрешностей устраняются программным путём в процессе обработки измерений.

4.1. Искажения, связанные с построчным сканированием

Процесс сканирования поверхности в сканирующем зондовом микроскопе происходит в условиях присутствия позиционного шума пьезопривода, который проявляется в случайных отклонениях измерений рельефа в соседних строках. Для того чтобы уменьшить влияние шумов в процессе сканирования, применяют фильтрацию изображения, устраняющую шум (например,

см. [22]). В микроскопе производства НПО «АЛМАЗ» [23] в каждой точке измерения проводятся несколько раз подряд. Затем крайние значения отбрасываются, а оставшиеся усредняются.

Подобный метод не препятствует, однако, сохранению некоторого отличия измерений в соседних элементах соседних строк скана. В результате этого на изображении могут возникать достаточно резкие перепады в уровне яркости между близкими элементами в разных строках.

Кроме того, часто во время сканирования неопределённым образом изменяется расстояние между зондом и образцом. Это происходит вследствие микродвижений в элементах конструкции измерительной головки микроскопа или вследствие изменения состояния рабочей части зонда (например, захвата кончиком зонда наночастицы с поверхности образца). В результате на СЗМ изображении появляются ступеньки (рис. 1.12 а). В работе [1] предлагается следующий способ исправления таких дефектов можно с помощью процедуры выравнивания измерений рельефа по строкам.

В каждой строке сканирования находится среднее значение рельефа:

$$Z_j = \frac{1}{N} \sum_i Z_{ij},$$

где N – число точек измерения в строке. Затем из значений в каждой строке набора измерений вычитаются соответствующие средние значения: $Z_{ij}^o = Z_{ij} - Z_j$. В новом наборе в каждой строке среднее равно нулю. Поэтому в нем не содержатся ступеньки, связанные с резкими изменениями среднего значения в строках. Возникающие при этом межстрочные различия предлагается удалить фильтрацией сигнала по столбцам кадра. На рис. 1.12 б представлен результат выравнивания строк реального изображения поверхности, полученного с помощью атомного силового микроскопа (АСМ), которое содержит ступенчатое изменение яркости.

У данного способа фильтрации есть и очевидный недостаток. Наличие большого объекта на поверхности («выступа» или «углубления») обязательно приводит к нестандартной средней величине измерений в содержащих его

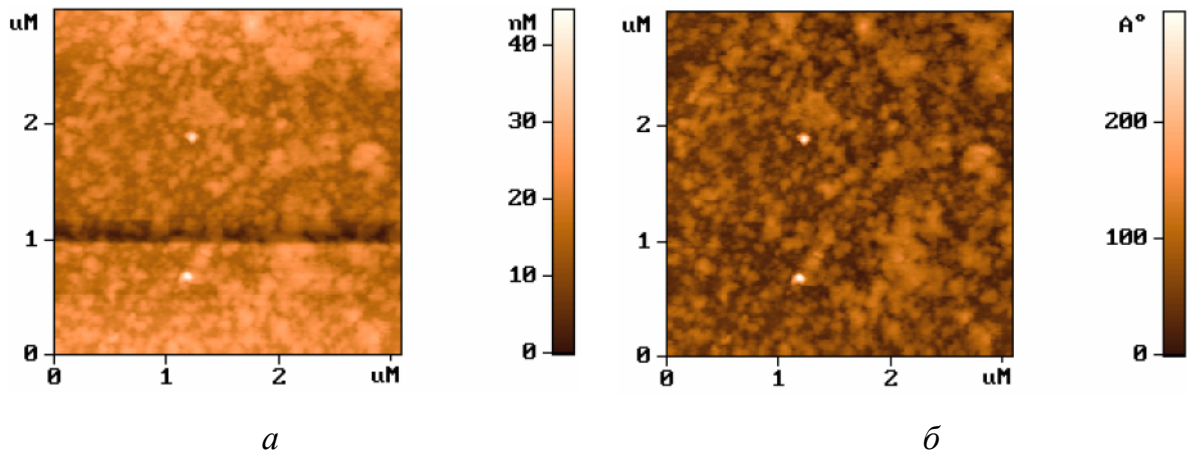


Рис. 1.12. АСМ изображения поверхности образца.

строках и вносит скачок в описание рельефа по столбцам сигнала на граничных строках таких объектов.

Для учета этого обстоятельства в цитируемой работе предлагается вычислять сумму измерений не по всем строкам для каждого столбца, а лишь по нескольким предшествующим строкам. Удовлетворительные результаты получаются при изменении яркости текущей строки, основываясь на среднем измерений 4-х предыдущих строк:

$$\begin{cases} Z'_{ij} = Z_{ij} - Z_j, \\ Z_j = \frac{1}{4} \sum_{k=i-4}^i Z_{kj}. \end{cases}$$

Заметим, что выполненные автором исследования изображений туннельного микроскопа производства НПО «АЛМАЗ», показывают, что для осреднения предпочтительно брать нечетное количество строк, так как отклонения от среднего уровня измерений в строке, как показывает сравнение соседних строк, обычно чередуются через строку. При дальнейшей обработке таких измерений следует учитывать то обстоятельство, что междустрочное осреднение не позволяет полностью избежать отмеченное чередование яркостей, а только уменьшает его.

4.2. Искажения модели рельефа, связанные с размерами иглы

В зондовой микроскопии за рельеф исследуемой поверхности принимается некоторая функция координат точек траектории движения острия иг-

лы в процессе сканирования. Необходимыми условиями того, чтобы реальная поверхность подложки совпадала с измеряемой, являются пренебрежимо малый по сравнению с точностью позиционирования диаметр конца иглы. На практике острое иглы имеет конечные размеры, поэтому получить реальное изображение рельефа поверхности подложки можно только, если знать геометрию острия. Учет реальных параметров формы иглы с острием конечного размера позволяет одновременно достичь двух целей: увеличить разрешающую способность зондового микроскопа и получить истинные соотношения между размерами наблюдаемых объектов в поле зрения.

Основой для применяемых в атомно-силовой микроскопии алгоритмов использования знания параметров геометрии острия иглы является метод, изложенный в работе [24]. В ней предлагается решение для случая плоской модели иглы и рельефа под ней в предположениях, что известны траектория движения зонда и, главное, - форма острия его иглы. На рис. 1.13 *а* они обозначены через I и T'' . В этом случае если зонд движется, касаясь иглой точек исследуемого рельефа, микроскоп измеряет и записывает параметры траектории I движения точки острия иглы. В работе показано, что кривая искомого рельефа S (см. рис. 1.4 *а*) является кривой, которую опишет конец острия перевернутой иглы при её движении с касанием снизу измеренной линии движения острия иглы зонда (см. рис. 1.13 *б*).

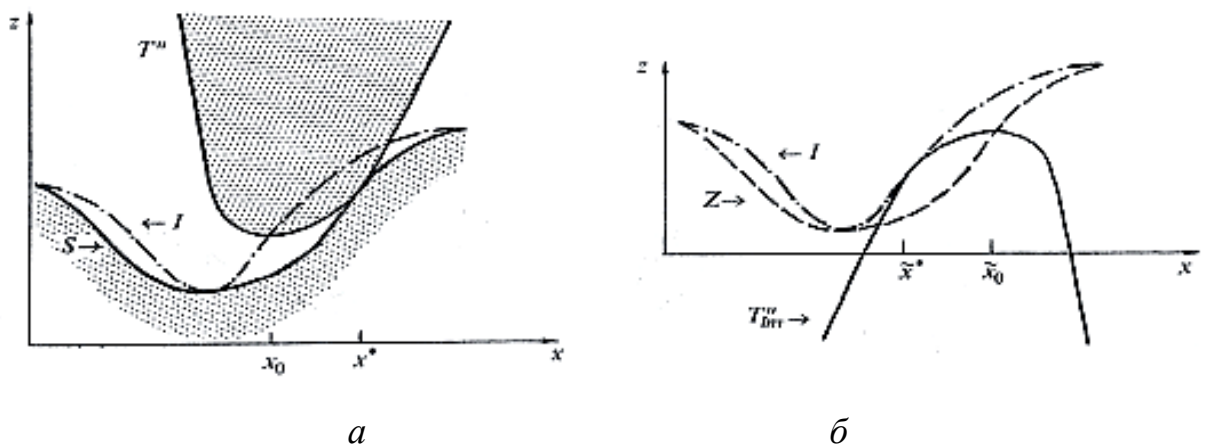


Рис. 1.13. Метод расчета рельефа поверхности

К сожалению, применение этого изящного с математической точки зрения метода решения задачи восстановления формы нанорельефа на практике затруднено отсутствием знания формы острия иглы. Кроме того, в реальных измерениях шумы измерения траектории I приводят к разбросу положений перевернутого зонда, и, поскольку по условию перевернутый зонд и траектория его движения имеют общую касательную, шумы знания движения зонда делают положение точки касания неопределённой.

В случае туннельной микроскопии появляется дополнительное ограничение, связанное с тем, что зонд не находится в контакте с поверхностью. Туннельный зазор варьируется в виду ошибок работы следящей системы и упомянутого окисления поверхности образца. Это приводит к ошибкам определения модельной точки контакта иглы с поверхностью исследуемого образца.

Таким образом, следует заключить, что практически все описанные выше методы построения рельефа исследуемого образца не гарантируют его наблюдаемость, поскольку они обладают повышенной чувствительностью к ошибкам модели иглы сканера и к шумам измерений.

4.3 Проблема определения формы иглы зонда микроскопа

Основанием для исследований решения задачи определения формы острия зонда является важный факт: пьезосканеры обеспечивают точность перемещения зонда на порядок лучше реальных размеров диаметра его иглы. Это позволяет рассчитывать на увеличение до 10 раз точности определения нанорельефа на том же оборудовании. Решению этой задачи препятствует трудности измерения формы конца иглы с нанометровым разрешением.

Различным подходам к решению задачи определения формы острия зонда и восстановления рельефа поверхности образца посвящено достаточно много работ [25–29].

В упомянутой работе [24] предложен способ, который приводит к решению задачи, если известен рельеф некоторой калибровочной поверхности и результат ее сканирования. Для нахождения формы и размеров острия иглы

предлагается из полученных измерений рельефа калибровочной поверхности вычесть её истинный рельеф. Приведенные результаты компьютерного моделирования работы алгоритма для нескольких примеров демонстрируют хорошее совпадение реальной и расчетной формы зонда. Этот способ широко используется в атомно-силовой микроскопии (рис. 1.14), где уровень достижимой точности позволяет использовать калибровочные поверхности с микронными размерами рельефа (которые можно создать и измерить). К сожалению, в туннельной микроскопии с её более высоким уровнем точности ка-

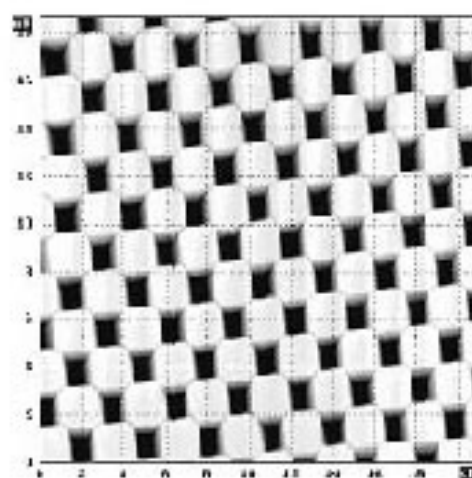
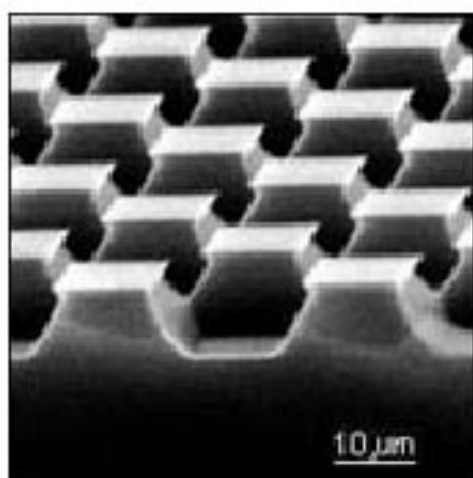


Рис. 1.14. Калибровочные поверхности для определения формы острия иглы при радиусе закругления 30 нм.

либровочные рельефы отсутствуют, поскольку технологические возможности современной техники не позволяют создавать на поверхности подложки тестовые элементы с характерным размером менее 30 нанометров.

Хорошее представление о состоянии проблемы восстановления поверхностей с нетривиальной структурой можно получить из работ [30–32]. Основные усилия в них направлены на поиск параметров формы иглы в случае, когда радиус её закругления не превышает десятков нанометров, и способы, основанные на использовании калибровочной поверхности, не применимы. При таких размерах иглы зонда возможности прямых измерений (например, с помощью электронной микроскопии) ограничены, в виду неэффективности их использования в массовом производстве игл. Кроме того, такой

метод не позволяет после установки определить ориентацию острия иглы относительно основания кантилевера, а также - учесть изменение ориентации иглы при его прогибе под действием силы тяжести и/или в силу внутренних деформаций.

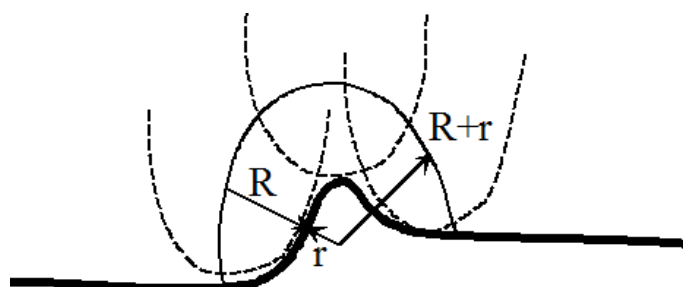


Рис. 1.15. Зависимость видимого радиуса закругления нанообразования от его диаметра $2r$ и радиуса закругления иглы R .

Конструктивный метод определения радиуса закругления для острых игл предложен в работе [31]. В нем предлагается использовать поверхности со сверхмалыми объектами, диаметр которых существенно меньше радиуса закругления острия иглы. Поверхность сканируют с помощью зондового микроскопа с исследуемой иглой. Для оценки радиуса закругления иглы на изображении выбирают мелкий объект. Он имеет вид выступа. В качестве диаметра острия принимают видимый диаметр такого объекта (рис. 1.15).

Недостатком рассматриваемого способа является отсутствие в нём возможности определения формы острия иглы. Кроме того, нетрудно показать, что видимый диаметр выступа пропорционален сумме диаметров объекта и острия. Следовательно, диаметр острия определяется с точностью до ширины выступа, который заранее неизвестен.

Главная причина практической ненаблюдаемости нанорельефа методами зондовой микроскопии состоит в отсутствии способа проверки модели острия зонда с требуемой точностью. Как было отмечено выше, это обстоятельство особенно болезненно для туннельной микроскопии, где достижимая точность движения иглы на порядок превышает знание её размеров.

По этой причине ниже главное внимание уделено исследованию и разработке программных средств для учета формы и размеров острия иглы применительно к задаче интерпретации измерений рельефа поверхности, полученных с помощью туннельного микроскопа.

Выводы первой главы

Отдавая должное простоте реализации и наглядности модели управления преобразованием данных, основанной на применении фильтров, следует заметить, что в нее не укладывается класс способов обработки данных, в которых результат не может быть получен в виде явного решения уравнения взаимодействия иглы зондового микроскопа с объектом измерения. К ним относятся случаи, когда геометрия острия иглы неизвестна либо физические условия взаимодействия могут изменяться в процессе измерения, например, на поверхности допускается наличие «адсорбата» (например, см. [33; 34]), либо учитываются особенности работы пьезопроводов движения иглы.

1. Зондовые микроскопы комплектуются программным обеспечением, которое позволяет управлять процессом сканирования и обрабатывать получаемые результаты. Проведенное сравнение позволяет сделать вывод о том, что их функциональные возможности в целом совпадают. Рассмотренные программные комплексы содержат модули первичной обработки измерений, которые позволяют уменьшить уровень шумов, модули двухмерной и трехмерной визуализации измерений, модули преобразования изображений, которые улучшают видимость деталей рельефа, и модули структурного и количественного анализа формы рельефа.
2. Все известные программные комплексы используют схожие программные модели. Они построены в виде монитора пользовательских заданий на обработку массивов данных с измерениями. Задание представляет собой команду, определяемую пунктом оконного меню, который фиксирует способы в последовательности преобразований данных: вызов и сохранение

файла с данными, применение фильтров и функций к данным, отображение на дисплее в соответствии с указанными пользователем параметрами и т.д.

3. Описанные общепринятые средства повышения визуального разрешения картины измерений путём учета наклона и средней кривизны поверхности приводят к результату, который нельзя признать удовлетворительным, так как получаемое изображение искажает геометрию рельефа. Это служит основанием для поиска способов развития применяемых методов.
4. Программные методы коррекции получаемых измерений рельефа с учетом влияния геометрии конца иглы дают неплохие результаты только для игл со сравнительно большим радиусом закругления их острия (не менее 30 нм). Это ограничивает наблюдаемость размеров нанорельефа и точность интерпретации его параметров аналогичными порядками величин.
5. В настоящее время для сверхострых игл зондовых микроскопов предложен простейший способ косвенной оценки радиуса закругления их острия с точностью до неизвестной величины аддитивного члена.
6. Для туннельной микроскопии и атомно-силовой микроскопии высокого разрешения имеется необходимость дополнения программных средств интерпретации измерений алгоритмами учёта влияния формы острия иглы зонда на процесс формирования изображения.

Глава 2. АЛГОРИТМ ОПРЕДЕЛЕНИЯ ГЕОМЕТРИИ ОСТРИЯ ИГЛЫ

В туннельном микроскопе рельеф поверхности измеряется с помощью зонда – иглы, острие которой имеет радиус закругления порядка нескольких нанометров. В процессе измерений запоминаются положения, которые зонд занимал при движении вдоль линии сканирования. Совокупность выполненных измерений принимается в качестве рельефа исследуемой поверхности. Необходимым условием того, чтобы результат интерпретации измерений совпадал с рельефом реальной поверхности сканируемого объекта ("подложки") является либо пренебрежимо малый диаметр острия иглы, либо знание его размеров и формы.

Восстановление реальной формы рельефа является достаточно сложной задачей учёта физических факторов, влияющих на туннельный ток и параметры работы системы управления пьезопроводов. В общей постановке задача сводится к решению уравнений в частных производных к позиции зонда, где неизвестными является граничные условия в виде поверхности иглы и искомого рельефа. Существующая математическая сложность прямого решения подобной задачи приводит к необходимости поиска её приближенных решений, которые лишь частично учитывают действующие факторы.

В этой главе исследуется влияние размеров конца иглы и физики туннельного тока на получаемое изображение поверхности. Результаты исследования приводят к выводу о необходимости поиска алгоритма коррекции изображения сигнала в тех случаях, когда требуется получить разрешение 1 нм и менее. Конец главы посвящен описанию алгоритма определения геометрии острия иглы путем интерпретации измерений рельефа поверхности с учетом физической модели процесса сканирования. Предполагается, что известны измерения рельефа, выполненные с помощью исследуемой иглы, и параметры сканирования, которые позволяют с помощью существующих моделей туннельного тока вычислить величину туннельного зазора.

§1. Принцип работы туннельного микроскопа

1.1. Модель туннельного тока

Основным способом получения рельефа поверхности с помощью СТМ является её сканирование с поддержанием постоянной величины туннельного тока. В этом режиме на иглу микроскопа подается постоянное напряжение, и система управления поддерживает такую высоту до поверхности, чтобы величина туннельного тока оставалась постоянной. Если поверхность является однородной, и между иглой и подложкой нет других веществ, то в отсутствие возмущений рельеф поверхности отслеживается концом иглы.

Неточности процесса отслеживания туннельного тока и особенности его протекания приводят к тому, что расстояние между острием иглы и подложкой не является постоянной величиной. В результате траектория движения конца иглы может отличаться от программной, и, вследствие этого, измеренная траектория движения зонда микроскопа отличается от реального рельефа поверхности подложки [23].

Зависимость плотности туннельного тока от расстояния до подложки дается формулой [1]:

$$j = \frac{e\phi^*}{2\pi\hbar z^2} \left(\exp(-kz) - \left(1 + \frac{eV}{\phi^*}\right) \exp\left(-k\sqrt{1 + \frac{eV}{\phi^*}}z\right) \right),$$

$$k = \frac{4\pi\sqrt{2m\phi^*}}{\hbar}.$$

Здесь приняты следующие обозначения действующих факторов:

j - плотность туннельного тока [$A \cdot m^{-2}$],

k - константа затухания волновой функции электрона, движущегося в области потенциального одномерного барьера прямоугольной формы [m^{-1}],

$m = 9,1 \cdot 10^{-31} \text{ кг}$ - масса электрона,

$e = 1,6 \cdot 10^{-19} \text{ Кл}$ - заряд электрона,

$\phi^* \approx 4 \text{ эВ} = 6,4 \cdot 10^{-19} \text{ Дж}$ - работа выхода электрона,

$h = 6,6 \cdot 10^{-34}$ Дж·с - постоянная Планка,

V - подводимое напряжение [В],

z – расстояние до поверхности [м].

При указанных значениях констант, k примерно равно $2 \cdot 10^{10} \text{ м}^{-1} = 2 \text{ \AA}^{-1}$.

В процессе сканирования подводимое напряжение меньше 1 В. Следовательно, $eV < 1,6 \cdot 10^{-19} \text{ Дж} < \phi^*$.

Используем эту зависимость для упрощения формулы плотности туннельного тока, линеаризовав его по параметру $\frac{eV}{\phi^*} < 0,25$. В итоге получим:

$$j = \frac{e^2 k V}{4 \pi \hbar z} \exp(-kz).$$

Подставив типичное значение подводимого напряжения $V = 0,3 \text{ В}$, находим, что $j = \frac{0,45 \cdot 10^{-6} k}{z} \exp(-kz) [A/m^2]$. Эту формулу удобно исследовать, если расстояние до поверхности z представить измерять в ангстремах ($\text{\AA}$). Тогда

$$j = \frac{0,9 \cdot 10^{-6}}{z} 7^{-z} [A/\text{\AA}^2].$$

В этой формуле z является безразмерной величиной, численно равной туннельному зазору, выраженному в ангстремах. Диаметр атома приблизительно равен $2 \text{\AA}$. Если в процессе сканирования расстояние между иглой и подложкой больше диаметра атома, т.е.

$$z > 2 [\text{\AA}],$$

то

$$7^{-z} \ll 1/z. \quad (2.1)$$

Из этого неравенства можно заключить, что зависимость туннельного тока от расстояния определяется исключительно скоростью убывания показательной функции (рис. 2.1). Удаление иглы от поверхности на $1 \text{\AA}$ приводит

к уменьшению туннельного тока в семь раз, т.е. почти на порядок (см. также [35]). Расстояние между атомами составляет порядка $3 \text{ \AA}$. Отсюда можно сделать вывод, что основная часть туннельного тока течет через ближайший к

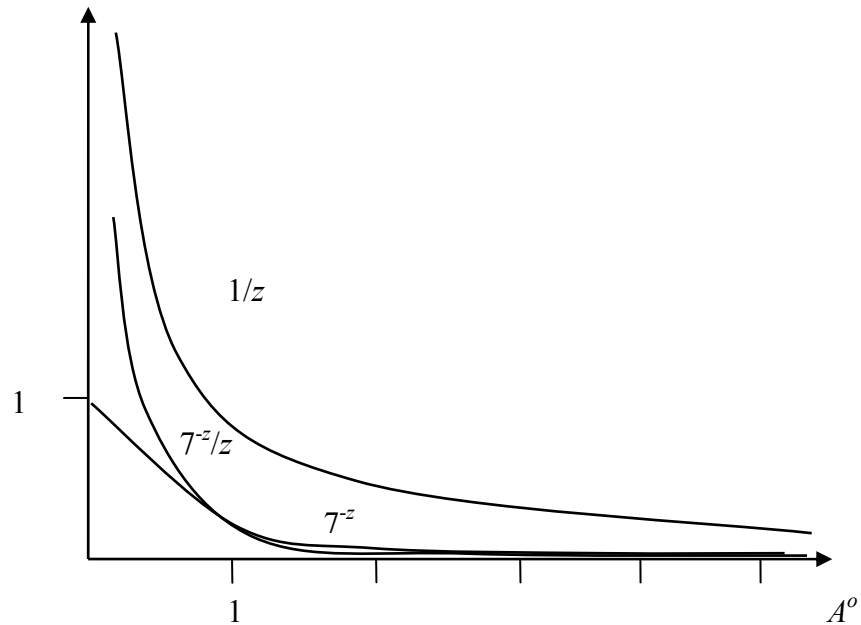


Рис.2.1. Графики функций $1/z$, 7^{-z} и $7^{-z}/z$.

сканируемой поверхности слой атомов, так как величина туннельного тока для второго слоя меньше в $7^{-3} = 350$ раз. Для того, чтобы существенная доля туннельного тока приходилась на второй слой атомов, форма конца иглы должна быть такой, чтобы диаметр второго слоя был бы в $\sqrt{350} \approx 18$ раз больше, чем первого.

Те участки острия, которые удовлетворяют этому условию, должны иметь большой радиус закругления хотя бы вдоль одного направления, т.е. быть почти прямолинейными. Этому условию, например, удовлетворяет острие с боковой поверхностью в виде конуса.

Рабочая величина туннельного тока равняется примерно $1 \text{ нА} = 10^{-9} \text{ А}$. Обозначив через S площадку на игле, через которую протекает большая часть тока, получим оценку для расстояния от этой площадки до подложки z , которая должно удовлетворять условию

$$\frac{S}{z} 7^{-z} = 10^{-3}. \quad (2.2)$$

Если условно принять, что площадь S равна площади атома ($4 \text{ \AA}^2$), то из этого неравенства следует, что расстояние, на котором должен двигаться кончик иглы для того, чтобы туннельный ток оказался равным 10^{-9} А , составляет $4 \text{ \AA}$. В литературе (в том числе в [1]) гипотетически считается возможной такая ситуация, когда весь туннельный ток течет через атом, находящийся на конце иглы.

Рассмотрим зависимость туннельного зазора от площади туннельного контакта S . Прологарифмировав равенство (2), после преобразований получим:

$$z = \frac{1}{\lg 7} (3 + \lg \frac{S}{z}) \cong 4 + 1,2 \lg S - 1,2 \lg z [\text{ \AA}]. \quad (2.3)$$

В силу неравенства (2.1), третий член можно считать малой константой. Это утверждение верно для всех

$$z < 10^{\frac{4}{1,2}} \approx 1000 [\text{ \AA}].$$

Например, при $z=4 \text{ \AA}$ $1,2 \lg z = 0,7$.

Из формулы (2.3) следует, что величина туннельного зазора мало меняется при изменениях площади туннельного контакта S . Действительно, для того, чтобы при постоянной плотности тока туннельный зазор увеличился на $1 \text{ \AA}$, площадь туннельного контакта S должна увеличиться в $10^{\frac{1}{1,2}} \approx 7$ раз. Нетрудно показать, что площадь туннельного контакта существенно увеличивается только при сканировании плоского участка поверхности боковой поверхностью острия, имеющего вид конуса. Это имеет место при сканировании наклонных участков рельефа. Однако вследствие экспоненциальной зависимости туннельного тока от туннельного зазора эти изменения будут компенсированы системой управления увеличением туннельного зазора всего на $1 \div 2 \text{ \AA}$.

Это дает основания в дальнейшем считать, что для игл с радиусом закругления острия порядка 1 нм туннельный зазор с точностью до $1 \text{ \AA}$ равен 5

$\text{\AA}$ (0,5 нм). Можно также говорить об относительной стабильности туннельного зазора при сканировании.

1.2. Взаимодействие иглы с поверхностью

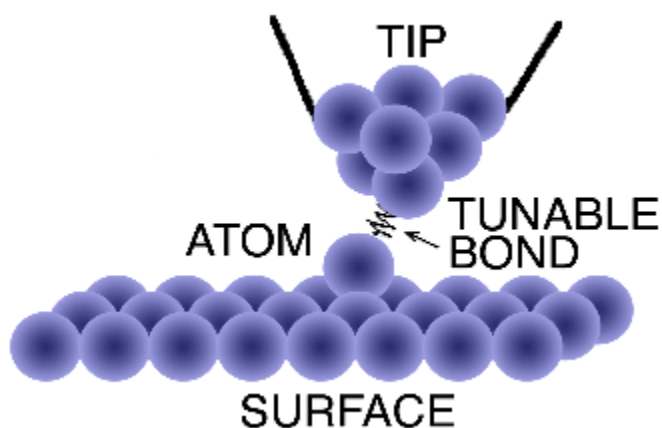


Рис. 2.2 Форма иглы и атом, через который проходит туннельный ток [36].

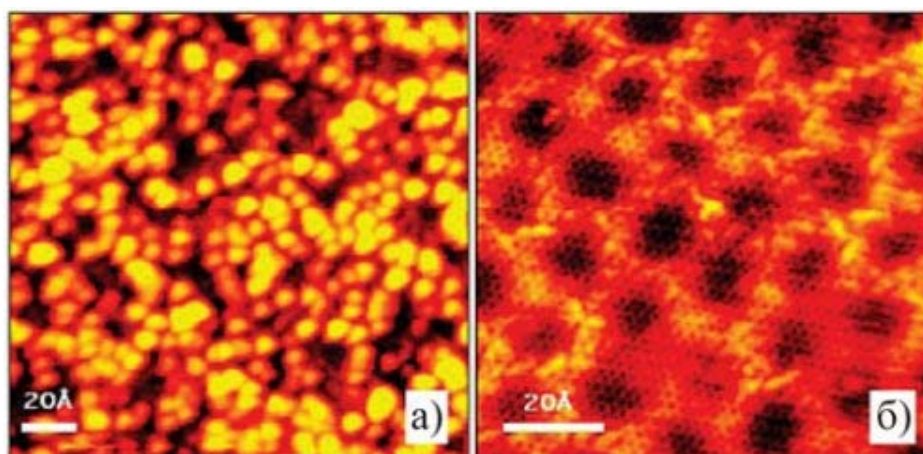


Рис. 2.3. **а** – СТМ изображение поверхности графена [37] после длительного воздействия источника водорода (90 с). Яркие светящиеся участки соответствуют водородным кластерным образованиям. **б** – Область графена, восстановленная после гидрирования посредством отжига при 800 °С.

Согласно существующим представлениям о природе туннельного тока [1; 38], весь ток проходит через один атом, который наиболее близко расположен к поверхности (рис. 2.2). Эти представления основываются на экспоненциальной зависимости величины тока от расстояния и тем, что с помощью туннельного микроскопа можно наблюдать атомную решетку атомарно гладких (специально подготовленных) поверхностей (рис. 2.3).

В общем случае доля ровных участков незначительна. Поэтому можно предположить, что при сканировании неровностей туннельный ток будет проходить через разные атомы поверхности острия [39]. Следствием этого является зависимость получаемого изображения от формы и размеров острия иглы.

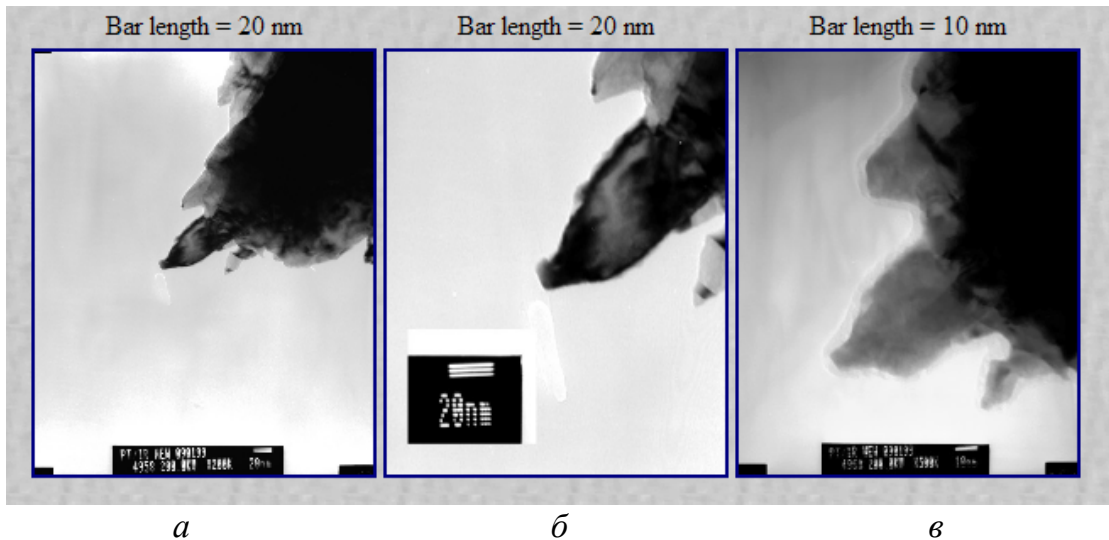


Рис. 2.4. Фотографии иглы СТМ с различным разрешением, полученные с помощью электронного микроскопа. Масштаб изображения указывает длина отрезка.
а, б – 20 нм, в – 10 нм (см. [40; 41]).

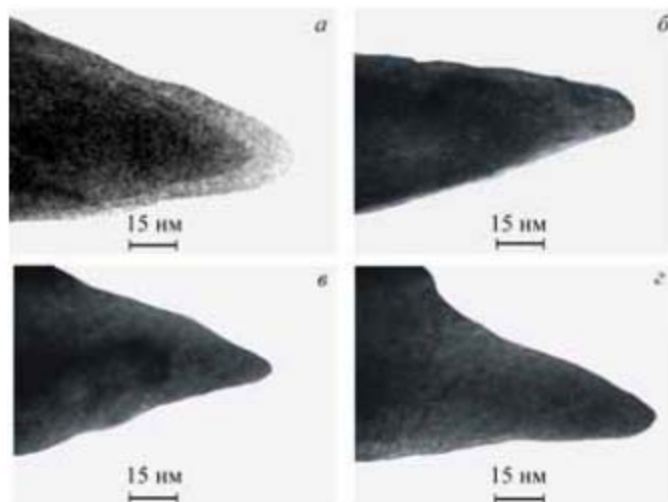


Рис. 2.5. Форма острия иглы на разных этапах изготовления.
а – после электрохимического травления [42], б – после удаления окисла ионным травлением, в – после 150 мин. ионной заточки, г – после 200 мин. ионной заточки.

На рис. 2.4 приведено несколько снимков зондов с помощью РЭМ, а на рис. 2.5 показаны изображения острий, полученных с помощью разных технологий.

Аналогичная точка зрения находит подтверждение в автореферате докторской диссертации Е.Ю. Шелковникова [43]. В нем приведены формулы для расчета туннельного тока при разных формах острия иглы и показаны области, через которые проходит туннельный ток.

Рассмотрим теперь влияние размеров конца иглы на получаемое изображение. На рис. 2.6 – 2.8 приведены примеры огибания иглой различных элементов рельефа.

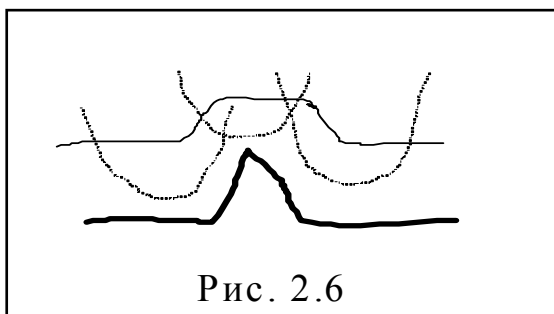
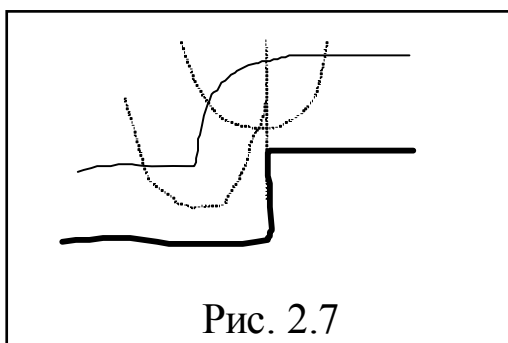


Рис. 2.6 соответствует огибанию объекта, имеющего вид пирамиды. Пример этого рисунка показывает, что:

- ширина изображения рельефа больше пирамиды на ширину иглы и удвоенную величину

туннельного зазора;

- на изображении нет острого угла;
- относительная высота изображения пирамиды приблизительно равна истинной высоте пирамиды.



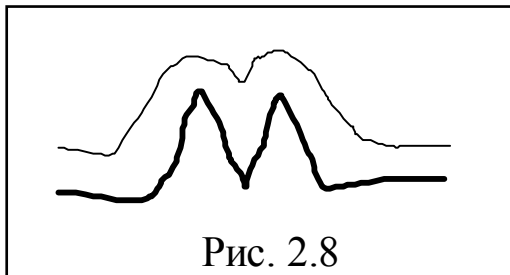
На рис. 2.7 рассмотрен случай огибания ступеньки [44]. Рисунок показывает следующее:

- на изображении уступ имеет вид дуги;
- эта дуга совпадает с половиной поверхности иглы, перевернутой концом

вверх;

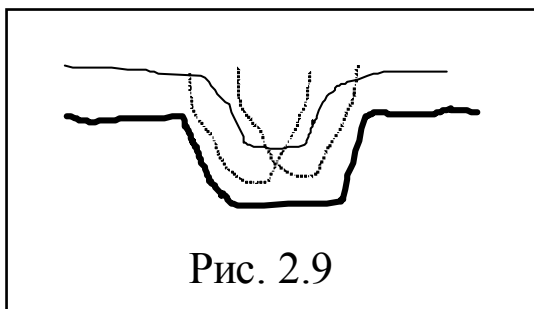
- высота уступа на изображении равна его истинной высоте;

- дуга сдвинута относительно уступа против направления сканирования на половину толщины иглы и туннельный зазор (суммарно 0,6 нм).



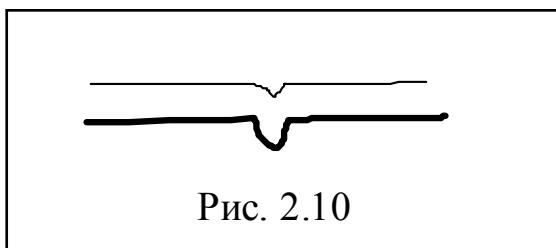
На рис. 2.8 изображен случай огибания двух пирамид, расстояние между которыми меньше размеров конца иглы. Из рисунка можно сделать вывод о том, что о наличии двух рядом расположенных пирамид

можно судить по наличию щели на изображении рельефа. Глубина щели на порядок меньше реальной. С учетом шумов определить наличие небольшой щели достаточно сложно.



На рис. 2.9 показано изображение впадины, которая шире конца иглы. Рисунок позволяет сделать следующие выводы:

- впадины на изображении уже на ширину иглы и удвоенный туннельный зазор;
- измеренная глубина впадины равна глубине реальной впадины только, если ее ширина больше ширины конца иглы и удвоенного туннельного зазора.



На рис. 2.10 изображен случай, когда ширина впадины меньше диаметра иглы. Поскольку туннельный ток является интегральной характеристикой проводящего слоя атомов, ширина и глубина впадины на изображении окажется во

много раз меньше, чем в реальности. Это позволяет сделать вывод о том, что без специальной программной обработки полученного изображения сигнала сложно увидеть такие детали, размер которых меньше не только «пятна проводимости», но и диаметра иглы.

§2. Алгоритм определения геометрии острия иглы туннельного микроскопа

В настоящей работе форма и размеры острия иглы туннельного микроскопа определяются путем обработки измерений сканирования исследуемой поверхности. Кроме измерений высоты и горизонтальной позиции иглы сигнал туннельного микроскопа содержит также параметры управления сканированием, которые определяют физические условия взаимодействия иглы и поверхности подложки: напряжение туннельного зазора и туннельный ток. Эти параметры позволяют вычислить туннельный зазор с помощью методики, изложенной в параграфе 1.

Если предположить, что геометрия острия иглы известна, то рельеф сканируемой поверхности можно определить с помощью геометрических вычислений, использующих данные сканирования [45]. Обозначим через $M(G,P)$ модель рельефа поверхности, которая получается из данных сканирования при условии, что острие иглы имеет геометрию G , а P – параметры сканирования. Форма и размеры острия иглы и рельеф поверхности должны быть такими, чтобы в совокупности выполнялись условия сканирования:

$$\forall m (m=1, \dots, M) U_m (G, M(G,P), P, t) = true, \quad (2.4)$$

где U_m – одно из M условий сканирования, t – текущее время. В качестве условий сканирования U_m могут быть выбраны, например, совпадение измеренных и модельных значений физических параметров, выполнение критериев, которые описывают геометрию объектов на поверхности, и условия, которые характеризуют траекторию движения зонда. Приближенное решение этих уравнений можно получить, выбирая форму и размеры острия зонда, минимизирующие количество точек, в которых нарушается соотношение (2.4). Реализация такого способа ведет к необходимости перебора различных моделей геометрии острия иглы, настройке их параметров и оценке векторов невязок при всех j и t .

В предложенной модели алгоритмов программных средств туннельной микроскопии рельеф исследуемой поверхности $M(G)$ следует определять путем вычисления поверхности пространственной фигуры, которая является объединением внутренностей туннельных оболочек выбранной формы иглы G для всех точек, в которых находилась игла при сканировании:

$$M(G) = \bigcup_{i=1}^N \bigcup_{j=1}^K M_{ij}(G). \quad (2.5)$$

Здесь $i=1, \dots, N$ – номер строки сканирования, $j=1, \dots, K$ – номер точки в строке, в которой проводилось измерение, а $M_{ij}(G)$ – вычисляемая в точке измерения с индексами i и j туннельная оболочка иглы формы G (расширение последней на величину туннельного зазора при измеренной величине туннельного тока рис. 2.11).

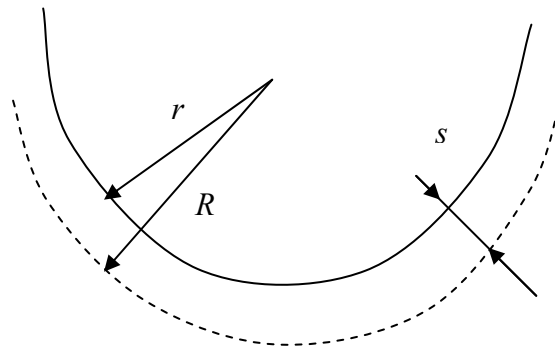


Рис. 2.11. Тунн

вывод оболочка (пунктирная линия) острия зонда (сплошная линия). r – радиус закругления острия, R – радиус закругления туннельной оболочки, d –

При решении любого уравнения из системы уравнений (2.4), из имеющихся моделей G формы иглы надо выбирать такую, для которой в наибольшем числе узлов k сетки измерений выполняется неравенство

$$\rho(M_k(G), M(G)) \leq \varepsilon, \quad (2.6)$$

где $\rho(M_k(G), M(G))$ – расстояние от туннельной оболочки k -й контрольной точки до поверхности рельефа $M(G)$, а ε – параметр алгоритма, определяющий допустимую величину отклонения туннельной оболочки от рельефа поверхности. Его величина обусловлена ошибками измерений и погрешностями вычислений. Выполнение неравенства (2.4) означает, что во всех контрольных точках условия взаимодействия зонда с поверхностью наилучшим образом соответствуют модели физических параметров сканирования, записанным в процессе измерения рельефа.

Критерий выполнения условий сканирования в каждом узле i, j измерения является локальным критерием, так как для его проверки не требуется знать геометрию всей области сканирования. Проверка локального критерия может быть выполнена сразу после вычисления очередной точки моделируемого рельефа поверхности $M(G)$.

Дополнительно к локальным критериям, могут быть предложены и глобальные критерии выполнения условий сканирования, которые учитывают особенности строения всего рельефа. Для проверки этих критериев должны быть вычислены все точки модели рельефа $M(G)$ и построены двумерная или трехмерная модели поверхности. Следовательно, проверка локальных и глобальных критериев выполнимости условий сканирования должна осуществляться в разных частях программной модели. Заметим, что в описываемом методе интерпретации сигнала туннельного микроскопа качество найденного решения фактически определяется не точностью получаемого результата обработки измерений (которая неизвестна и принципиально не может быть определена формальным способом), а предложенным способом количественной оценки степени выполнения физических условий процесса сканирования.

Для разных уравнений системы (2.4), их приближенные решения могут приводить к противоречивым результатам. Это существенно ограничивает возможности автоматизма программных средств и фактически делает необходимой интерактивную обработку данных с многокритериальным выбором решения (по числу условий U_m). В связи с этим потребовалась разработка

программной модели именно интерактивной реализации описанного выше метода непараметрической обработки данных сканирования. Важно отметить, что построенная модель программного комплекса (она описывается в следующей главе) может применяться для широкого класса задач нанотехнологии. Например, изменение модели физики взаимодействия иглы и сканируемой поверхности позволяет в рамках той же модели искать форму острия иглы в атомно-силовой микроскопии (как в контактном, так и бесконтактном режимах сканирования).

Выводы второй главы

1. Геометрия острия иглы существенно влияет на точность измерения формы и величины объектов нанометрового размера, причем при учете искажений следует принимать также величину туннельного зазора.
2. Предложенный алгоритм определения формы и размеров острия иглы состоит в подборе такого решения, для которого наиболее полно выполнены геометрические условия сканирования при заданных физических параметрах процесса измерения. При этом геометрия острия и рельеф поверхности определяются одновременно.
3. С точки зрения программной реализации особенностью алгоритма является наличие двух вложенных потоков интерактивного выбора: выбор параметров геометрии туннельной оболочки и выбора критерия для оценки соответствия полученного решения геометрическим условиям сканирования.

Глава 3. ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО КОМПЛЕКСА ДЛЯ ИНТЕРПРЕТАЦИИ ДАННЫХ СКАНИРОВАНИЯ

Реализация рассмотренного алгоритма связана с решением ряда взаимосвязанных задач:

- разработка программной модели процесса интерактивного поиска приближенного решения;
- обеспечение контроля за процессом решения путем визуализации результатов расчетов;
- обеспечение возможности усовершенствования геометрических и физических моделей с целью получения более точных решений;
- увеличение производительности вычислений за счет использования многоядерности/многопроцессорности современных вычислительных машин;
- снижение трудоемкости и сложности реализации перечисленных возможностей программного комплекса путем использования современных средств разработки.

В главе обосновываются решения перечисленных задач, которые использовались при создании программного комплекса для обработки данных сканирования рельефа поверхности с помощью туннельного микроскопа.

§1. Форматы входных и выходных данных

1.1. Форматы входных данных

В настоящее время в зондовой микроскопии не существует единого стандарта представления измерений. Поэтому большинство программных комплексов кроме собственного формата данных поддерживает ввод и вывод информации в достаточно большом числе специализированных и стандартных графических форматах.

В качестве основного входного формата данных в диссертационной работе выбран формат NWF. Он предназначен для программного комплекса StmShell и был создан Институтом нанотехнологий в г. Стирлинг (Шотландия) в середине 1990-х годов. Модификация данного программного комплекса используется в системном программном обеспечении ряда отечественных туннельных микроскопов, например, производства НПО «АЛМАЗ» и некоторых микроскопах производства концерна «Наноиндустрия». С помощью этих микроскопов проводилась большая часть исследований с использованием разработанного комплекса.

Помимо исходных значений измеренного сигнала, получаемого от туннельного микроскопа, в файл формата NWF записана метаданная. Она включает в себя ширину и длину сканированной области в пикселях и нанометрах, нижнюю и верхнюю высоты рельефа, физические условия сканирования, время начала сканирования и задержки первого измерения, а также комментарии пользователя и дополнительную информацию о системных параметрах. Информация о нижней и верхней высотах рельефа служит для масштабирования измерений по высоте.

Комментарии пользователя, единица измерений и некоторая дополнительная информация хранятся в строковом виде. Строки записаны в формате Pascal/Delphi. Формат подразумевает однобайтовое представление каждого символа. Первый байт строки хранит ее общую длину (без учета данного байта). Поэтому максимальная длина комментария составляет 255 символов.

Существенным для работы достоинством формата NWF явилось хранение измерений, полученных напрямую от туннельного микроскопа, и параметров физических условий сканирования. Каждое измерение хранится в виде знакового 16-разрядного слова со значением сигнала. Значение сигнала может изменяться от указанного в файле минимального значения сигнала до максимального. Заметим, что значение 16-разрядного слова, описывающего минимум сигнала, может быть отрицательным.

Таблица 3.1. Формат данных программного комплекса StmShell.

const int StaticPartSize = 0x4e	Данная константа задает размер первой части заголовка (не включая информацию о строках)
Int16 signature	Идентификатор версии формата файла (всегда равен 0x0003)
Int16 width_px	Количество измерений в строке
Int16 scanLineCount	Количество строк с измерениями
Int16 rawReliefMinimum_int16	Минимальное значение сигнала от АЦП туннельного микроскопа для данного образца
Int16 rawReliefMaximum_int16	Максимальное значение сигнала от АЦП
UInt16 rawReliefMinimumSign	Знак минимального значения сигнала от АЦП (0xffff, если отрицательное, иначе 0x0000)
UInt16 rawReliefMaximumSign	Знак максимального значения сигнала от АЦП (также 0xffff или 0x0000)
Int16 reserved1	Зарезервировано
Int32 firstMeasureDelay_microsec	Задержка перед первым измерением СТМ в микросекундах (обычно 0)
Int64 reserved2	Зарезервировано
Single width_nm	Ширина измеряемой части образца в нанометрах
Single length_nm	Длина измеряемой части образца в нанометрах
Single reliefMinimum_nm	Минимальная высота рельефа поверхности в нанометрах
Single reliefMaximum_nm	Максимальная высота рельефа поверхности в нанометрах
Single probeMovingSpeed_nm_s	Скорость перемещения зонда в нанометрах в секунду
byte[] signalLevel	Регулировка уровня сигнала (выше/ниже)
byte stringInfoLength	Длина в байтах строки информации

Таблица 3.1. Формат данных программного комплекса StmShell.

	о типе сканирования
byte[] stringInfo	Строка информации о типе сканирования (обычно «Fine Relief» или «Rough Relief»)
byte stringMeasureUnitLength	Длина в байтах строки с единицей измерения
byte[] stringMeasureUnit	Строка с единицей измерения (всегда «nm»)
const int DynamicPartSize = 0x27	Данная константа задает размер второй части заголовка (не включая информации о строках)
UInt32 startScanningTime	Время начала сканирования в формате UNIX time_t
Single tunnelFunction_mV	Параметры туннельной функции: напряжение в милливольтках
Single tunnelFunction_pA	Параметры туннельной функции: ток в пикоамперах
Single gateRangeBottom	Нижняя граница технологического диапазона изменения АЦП высоты в нанометрах
Single gateRangeTop	Верхняя граница технологического диапазона
Single PidPart1, PidPart2, Pid-Part3	Внутренний идентификатор
byte stringCommentLength	Длина в байтах строки комментария пользователя
byte[] stringComment	Строка информации комментария пользователя
Int16[] relief	Сигнал от АЦП туннельного микроскопа в точках измерения, записанный точка за точкой, строка за строкой

Для обеспечения совместимости с другими программными комплексами, разработанный комплекс позволяет получать информацию о сигнале, используя его изображение в формате BMP. Для работы с этим форматом тре-

буются файл с изображением, файл с палитрой, информация о длине или ширине изображения и перепад высот в нанометрах. Необходимые данные можно получить из имеющегося изображения, если на нем, как это часто бывает, отображена сопутствующая информация о масштабе, размерах скана и условиях сканирования.

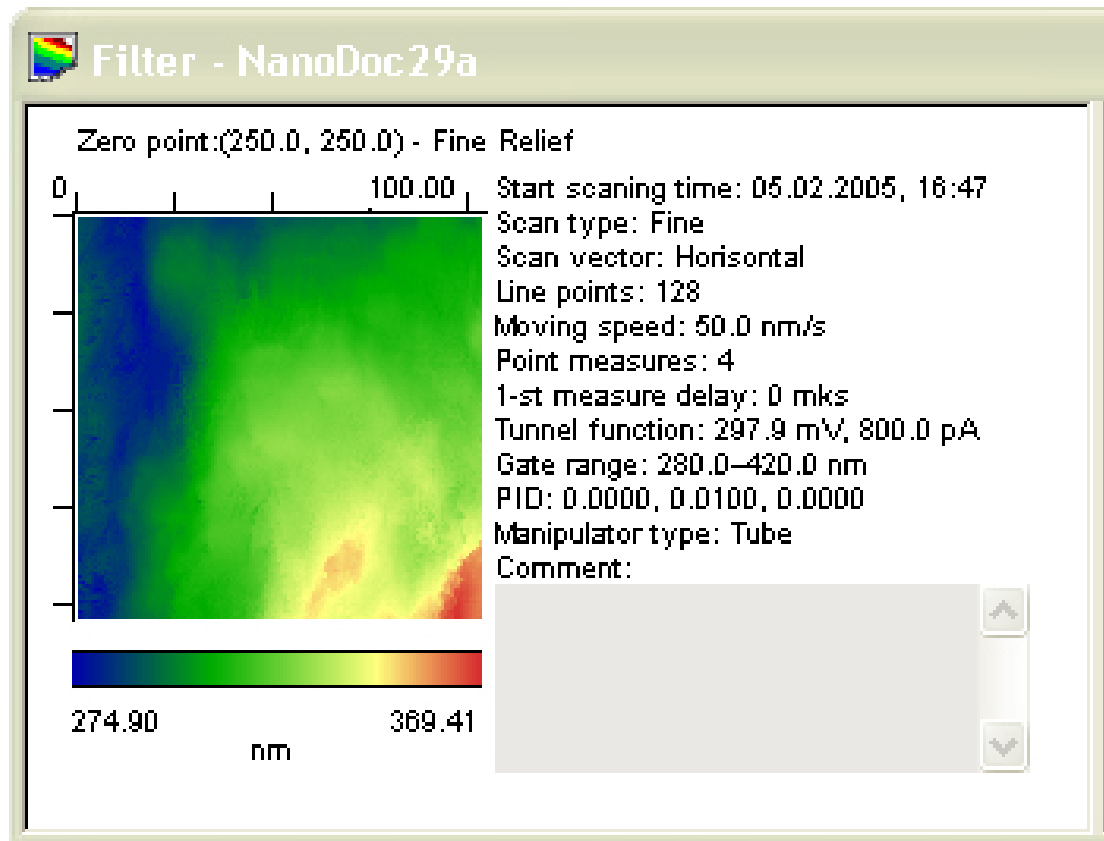


Рис. 3.1. Типичный вид окна с результатами измерения.

На рис 3.1 приведен пример результатов измерения в том виде, в котором он представляется пользователю. В левой части окна представлены результаты измерений, используемая палитра, в правой части – служебная информация об условиях измерения.

Достаточно часто изображения поступают без информации о физических условиях сканирования. В таком случае невозможно определить величину туннельного зазора. В главе 2 было показано, что следствием экспоненциальной зависимости величины туннельного тока от туннельного зазора является незначительные изменения туннельного зазора в абсолютном выражении в достаточно широком диапазоне изменения величин параметров скани-

рования. Поэтому в случае, когда все необходимые параметры сканирования недоступны, в качестве начального приближения можно принять величину туннельного зазора равной 0,5 нм. В программном комплексе также реализована возможность указать величину туннельного зазора выбором специального пункта меню.

Палитра должна представлять собой файл, в котором записаны пиксели с встречающимися в изображении сигнала цветами в линейной шкале. Горизонтальные координаты пиксела определяет высоту рельефа поверхности. Минимальная координата соответствует минимуму высоты, а вертикальная – максимуму. Интервалом изменения высоты назначается диапазон от нуля до максимальной координаты по горизонтали пиксела палитры. Вертикальные координаты пикселей значения не имеют. Таким способом в процессе преобразования данных палитра интерпретируется в набор «цвет-высота» («яркость-высота» для монохромных палитр).

Для каждого цвета пиксела входного изображения подбирается два цвета из палитры, максимально близких к текущему по евклидовой метрике в трехмерном пространстве цветовой модели RGB:

$$\rho(\{r^*, g^*, b^*\}, \{r_i, g_i, b_i\}) = \sqrt{(r_i - r^*)^2 + (g_i - g^*)^2 + (b_i - b^*)^2} \rightarrow \min_i,$$

где $\{r_i, g_i, b_i\}$ – цвета из палитры, а $\{r^*, g^*, b^*\}$ – цвет, для которого проводится палитризация. После этого высота определяется в соответствии с высотами, соответствующими выбранным двум пикселям, путем линейной интерполяции с учетом расстояния между цветом входного пиксела и двумя выбранными цветами:

$$h = x_1 + \frac{\rho(\{r^*, g^*, b^*\}, \{r_1, g_1, b_1\})}{\rho(\{r^*, g^*, b^*\}, \{r_2, g_2, b_2\})}(x_2 - x_1),$$

где $\{r_i, g_i, b_i\}$, $i=0;1$, – выбранные цвета из палитры, а x_1 и x_2 – горизонтальные координаты пикселей первого и второго цветов соответственно.

После ввода массива высот, осуществляется нормировка его элементов для того, чтобы сетка и величины высот имели одну единицу измерения.

Нетрудно заметить, что рассматриваемая схема позволяет определять высоту точки на изображении по ее цвету даже в случае, когда искомый цвет в палитре пропущен. Это имеет место всегда, когда количество цветов на изображении соизмеримо или даже больше, чем с количеством цветов в палитре. Например, при сохранении изображения, представленного на дисплее, количество цветов палитры ограничено длиной цветовой линейки и разрешением дисплея. Проблема обеспечения разрешающей способности в формате BMP подробно обсуждается ниже в разделе выводных форматов.

Вопрос разрешающей способности вводимой информации тесно связан с проблемой представления графической информации в файловых форматах. Они должны обеспечить не только представление данных с высоким разрешением, но и хранение метаданных, которая необходима для правильной интерпретации измерений. Отсутствие такой возможности у популярных графических форматов (BMP, GIF и др.) приводит к необходимости использовать для этой цели специализированные форматы хранения структурированных данных. Примером такого формата является XML. В отличие от бинарных форматов, XML содержит метаданные об именах, типах и классах описываемых объектов, по которым приложение может обработать документ неизвестной структуры. Преимуществом XML для зондовой микроскопии является возможность хранения координаты точки рельефа поверхности с высокой точностью. Это позволяет использовать формат, не беспокоясь о возможном ухудшении разрешающей способности по вертикальной координате. Кроме того, XML удобен для хранения метаданных о сканировании и о параметрах обработки данных. Недостатками данного формата являются необходимость использования трансформации XML-документа (XSLT) в случае, если программным комплексом используется неподдерживаемая по умолчанию схема документа, а также большой размер файла по причине хранения информации в текстовом виде.

1.2. Форматы выходных данных

Программный комплекс поддерживает следующие форматы выходных данных:

1. Файл высот

Описание формата высот приведено выше.

2. Файл высот в формате XML.

В файл этого формата записывается метainформация о процессе сканирования, а также список помеченных пользователем точек на рельефе и список высот рельефа поверхности образца, строка за строкой. Структура на языке XML Schema ([46]) генерируемого реализованным программным комплексом документа показана ниже.

Схемой описаны две структуры «parameters» и «scanLines», входящие в структуру «Relief», с их элементами. Элемент с адресом XPath /Relief/parameters/startScanningTime указывает дату и время начала сканирования в формате YYYY-MM-DDTHH:MM:SS (год-месяц-день, час:минута:секунда, например, 2006-05-02T11:33:12). Помеченные пользователем точки рельефа указываются как атрибуты подэлемента data элементов с адресом XPath /Relief/scanLines/ArrayOfPoint1d/Point1d. Чаще всего в программном комплексе встречается пометка точек цветом. В таком случае элемент data будет иметь вид

```
<data name="col" value="Color [Lime]" type="System.Drawing.Color" />.
```

Здесь Lime – требуемый цвет зеленого оттенка (лиметтовый, т.е. цвет лайма). В документе, построенном по данной структуре, высоты выводятся в единицах измерения, равных расстоянию между строками сканирования. Выбор такой метрики обусловлен тем, что расстояния между соседними отсчетами равны одному пикселу. Это гарантирует равенство масштабов по всем направлениям.

```

<xs:schema>
  <xs:import namespace="http://www.w3.org/2001/XMLSchema-instance"
    schemaLocation="xsi.xsd"/>
  <xs:element name="Relief">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="height_px" type="xs:decimal"/>
        <xs:element ref="ns1:parameters"/>
        <xs:element ref="ns1:scanLines"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element"/>
  <xs:element name="parameters">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="scanLineCount" type="xs:integer"/>
        <xs:element name="width_px" type="xs:integer"/>
        <xs:element name="length_nm" type="xs:integer"/>
        <xs:element name="width_nm" type="xs:integer"/>
        <xs:element name="rawReliefMinimum_int16" type="xs:integer"/>
        <xs:element name="rawReliefMaximum_int16" type="xs:integer"/>
        <xs:element name="rawReliefMinimumSign" type="xs:integer"/>
        <xs:element name="rawReliefMaximumSign" type="xs:integer"/>
        <xs:element name="firstMeasureDelay_microsec" type="xs:integer"/>
        <xs:element name="reliefMinimum_nm" type="xs:decimal"/>
        <xs:element name="reliefMaximum_nm" type="xs:decimal"/>
        <xs:element name="probeMovingSpeed_nm_s" type="xs:integer"/>
        <xs:element name="startScanningTime" type="xs:NMTOKEN"/>
        <xs:element name="tunnelFunction_mV" type="xs:decimal"/>
        <xs:element name="tunnelFunction_pA" type="xs:integer"/>
        <xs:element name="gateRangeBottom" type="xs:integer"/>
        <xs:element name="gateRangeTop" type="xs:integer"/>
        <xs:element name="info" type="xs:string"/>
        <xs:element name="measureUnit" type="xs:NCName"/>
      </xs:sequence>
      <xs:attribute ref="xsi:type" use="required"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="scanLines">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="ns1:ArrayOfPoint1d"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="ArrayOfPoint1d">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="ns1:Point1d"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Point1d">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" ref="ns1:data"/>
      </xs:sequence>
      <xs:attribute name="value" use="required" type="xs:decimal"/>
    </xs:complexType>
  </xs:element>

```

```

<xs:element name="data">
  <xs:complexType>
    <xs:attribute name="name" use="required" type="xs:NCName"/>
    <xs:attribute name="type" use="required" type="xs:NCName"/>
    <xs:attribute name="value" use="required"/>
  </xs:complexType>
</xs:element>
</xs:schema>

```

3. Файл в формате CSV.

Текстовый формат «comma-separated values» удобен своей простотой. Он предназначен для представления табличных данных. Каждая строка файла представляет собой одну строку таблицы. Значения отдельных колонок разделяются запятой или точкой с запятой. Данные в формате CSV легко просмотреть вручную, кроме того, ввиду своей простоты, он может быть открыт и отредактирован в большом количестве программ. В программном комплексе в файл этого формата выводятся значения высоты рельефа, строка за строкой. Количество колонок равно количеству координат рельефа в строке. Как и в случае XML, значения высот выводятся в условных единицах и также обеспечивают высокое разрешение, в отличие от графических форматов представления информации.

Фактически CSV представляет собой плоскую базу данных. Поэтому файл такого формата удобно подвергать анализу (например, проводить статистический анализ и рисовать диаграммы с помощью табличного процессора) и встраивать в экспертные системы. Текстовое представление данных в формате CSV позволяет предусмотреть в файле место для хранения метаданных.

4. BMP.

Формат хранения аппаратно-независимых растров в Microsoft Windows и IBM OS/2, более известный как BMP, позволяет выводить информацию о высотах рельефа поверхности образца в графическом виде. Для этого сохраняется изображение горизонтальной проекции рельефа, каждый пиксел которого окрашен в определенный цвет. Соответствие цветов и высот устанавливается с помощью палитры.

Из перечисленных форматов BMP наиболее удобен для использования, так как может быть прочитан всеми графическими редакторами, а также может быть встроен в текст в виде рисунка в редакторе WORD. Удобство формата состоит в том, что спектральный состав палитры может быть задан произвольным образом (например, черно-красная; серая; палитра, используемая в описанном выше программном комплексе StmShell с зелеными, красными, желтыми и синими оттенками и др.)

Однако при использовании формата BMP следует учитывать следующие обстоятельства. Часто рисунки BMP получаются пользователем путем копирования с экрана дисплея. Визуально они выглядят как исходные, но цветовой состав палитры оказывается не тождественно равным цветам, используемым операционной системой. Причинами этого является применение ИСС-профилей к изображению, находящемуся в кадровом буфере и ограничение глубины цветности рабочего стола. Дополнительную проблему представляет необходимость хранения информации о палитре наряду с самим изображением. При отсутствии палитры, восстановление ее вида по имеющемуся изображению также приводит к погрешностям при обратном преобразовании из изображения в массив высот.

Стоит отметить, что задача обеспечения соответствия «высота-цвет из палитры», вообще говоря, является некорректно поставленной, так как могут быть нарушены условия существования и однозначности ее решения:

- не для всех высот явно предопределен цвет в палитре;
- в палитре могут быть повторяющиеся цвета (либо по причине ее изначальной некорректности, либо как результат преобразований из-за дискретности машинного представления цвета).

Нарушение этих условий также приводит к неустойчивости обратного преобразования цвета в высоту: близким цветам могут быть назначены сильно различающиеся высоты.

Кроме того, способ сохранения метаинформации в файле формата BMP не стандартизован и является недокументированным. Поэтому единственным

корректным способом сохранения информации о масштабе, размерах рисунка, условиях и параметрах сканирования в файле этого формата является ее генерирование в виде изображения и добавление ее к имеющемуся изображению горизонтальной проекции рельефа.

§2. Особенности программной модели

Рассмотрим программную модель процессов обработки данных сканирования по алгоритму, изложенному в предыдущей главе (рис. 3.2). Модель включает в себя два модуля.

Модуль 1 содержит средства для обработки данных, путем моделирования взаимодействия зонда и поверхности при сканировании. Решение ищется пользователем путём интерактивного перебора заранее заданных моделей геометрии острия иглы и применения различных критериев для оценки результатов моделирования.

Пользователь вводит физические параметры, которые определяют условия сканирования, и задает параметры модели геометрии острия иглы. Затем запускается процесс вычисления модели рельефа поверхности. После того как модель рельефа построена, пользователь выбирает локальный критерий оценки выполнения физических условий сканирования и оценивает полученные результаты моделирования на соответствие ему, опираясь на представляемую программой графическую информацию. Решение должно удовлетворять всем локальным критериям во всех или, хотя бы, в наибольшем числе точек каркаса рельефа. Если построенная модель рельефа не удовлетворяет хотя бы одному локальному критерию, то параметры геометрии острия должны быть интерактивно уточнены пользователем.

Наличие блоков интерактивной обработки данных делает невозможным сведение описанной модели к описанной в первой главе традиционной модели программ обработки данных зондовой микроскопии, основанной только на их функциональных преобразованиях. Конечно, очевидное преимущество функционального метода состоит в объективности получаемого решения. Но

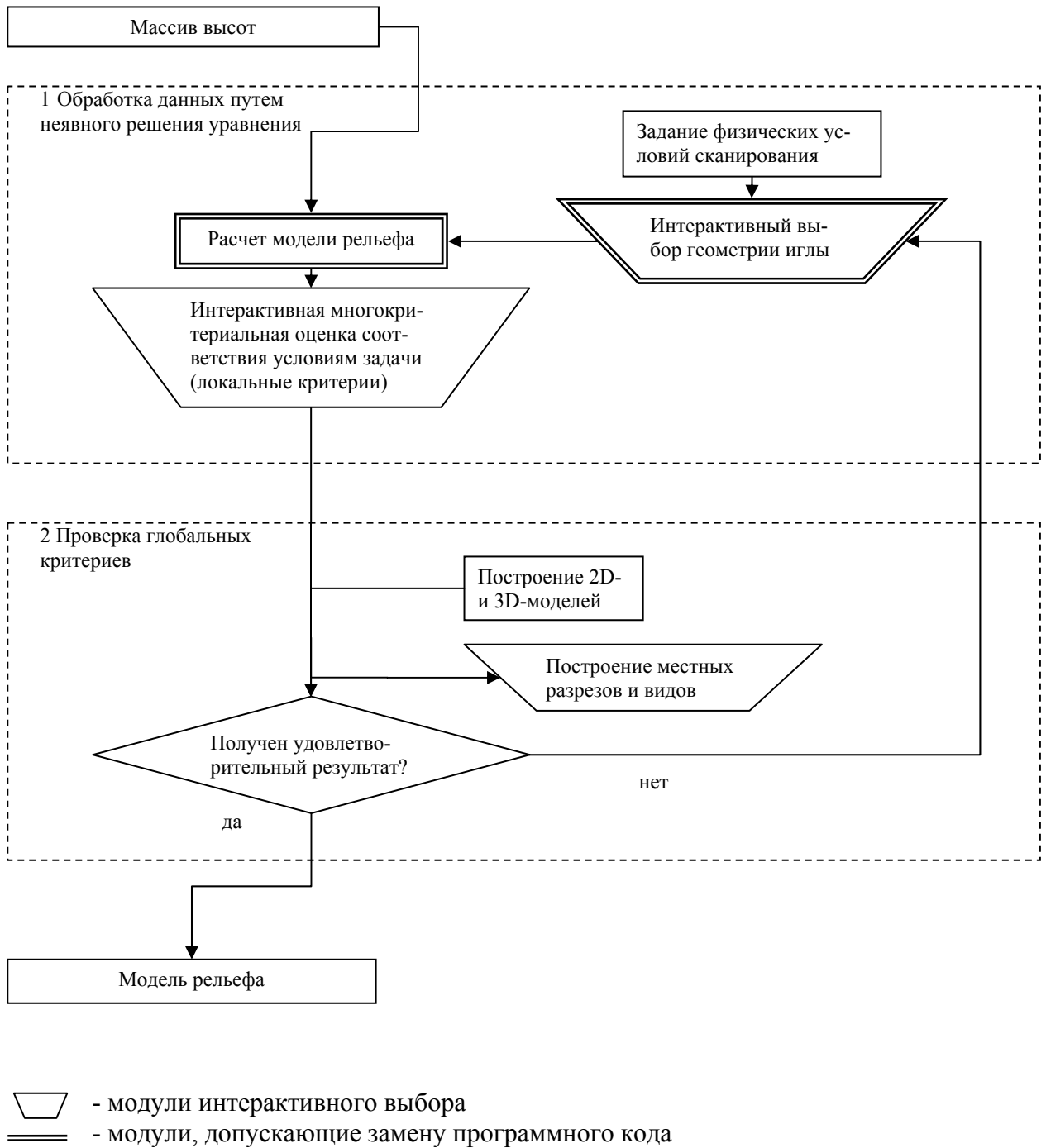


Рис. 3.2. Модель процессов обработки данных сканирования.

как в исследовательских задачах, так и в задачах, требующих точной настройки, это преимущество часто оказывается недостатком, так как в известной мере любое решение является приближенным и приводит к артефактам. Однозначность функционального подхода не позволяет пользователю оценить, что в получаемом результате является объективным свойством, а что —

артефактом, связанным с методом. Интерактивный метод делает возможным для пользователя исследование этого вопроса.

В модуле 2 содержатся средства визуализации рельефа поверхности для оценки пользователем правильности полученной модели нанорельефа подложки с учетом геометрических критериев. Обязательным условием обработки данных зондовой микроскопии является построение трехмерной модели рельефа, позволяющей пользователю оценить в целом построенную им картину рельефа. Также должно быть реализовано построение множества двумерных местных разрезов рельефа и видов под выбранным углом выбранной в интерактивном режиме части изображения рельефа поверхности. С помощью таких средств визуализации пользователь может контролировать соответствие области рельефа глобальным критериям физических условий сканирования. Если физические условия выполнены, то это означает, что выбранные характеристики острия игла могут быть приняты в качестве модели иглы, с помощью которой осуществлялось сканирование. При этом модель рельефа поверхности может быть принята в качестве поверхности, которая сканировалась.

Заметим, что предлагаемая модель обработки данных зондовой микроскопии имеет много общего с известными моделями графических редакторов. Например, в программах Adobe Photoshop или Paint выбор инструмента также осуществляется пользователем посредством интерактивной процедуры. В них также имеются средства для оценки результата преобразований. Отличие описываемой модели состоит в том, что в графических редакторах субъективная оценка изображения делается на основании только глобальных критериев (уровня яркости, цветовой гаммы изображения и др.), т.е. отсутствует предлагаемый интерактивный контур многокритериальной оценки соответствия результатов моделирования и измерений с использованием множества локальных критериев.

§3. Проектирование комплекса с заменяемыми модулями

Наиболее эффективным способом разработки программных средств является реализация их программного кода на основе метода объектно-ориентированного программирования [47]. Использование этого метода предполагает, что на стадии разработки структура программного комплекса известна или хотя бы хорошо проработана. Этому условию достаточно трудно удовлетворить при создании комплексов для исследовательских целей, так как в начале разработки состав функций и методов заранее известен лишь в общих чертах и должен быть определен в результате разработки. Поэтому структурирование программы оказывается возможным лишь после того, как комплекс фактически создан.

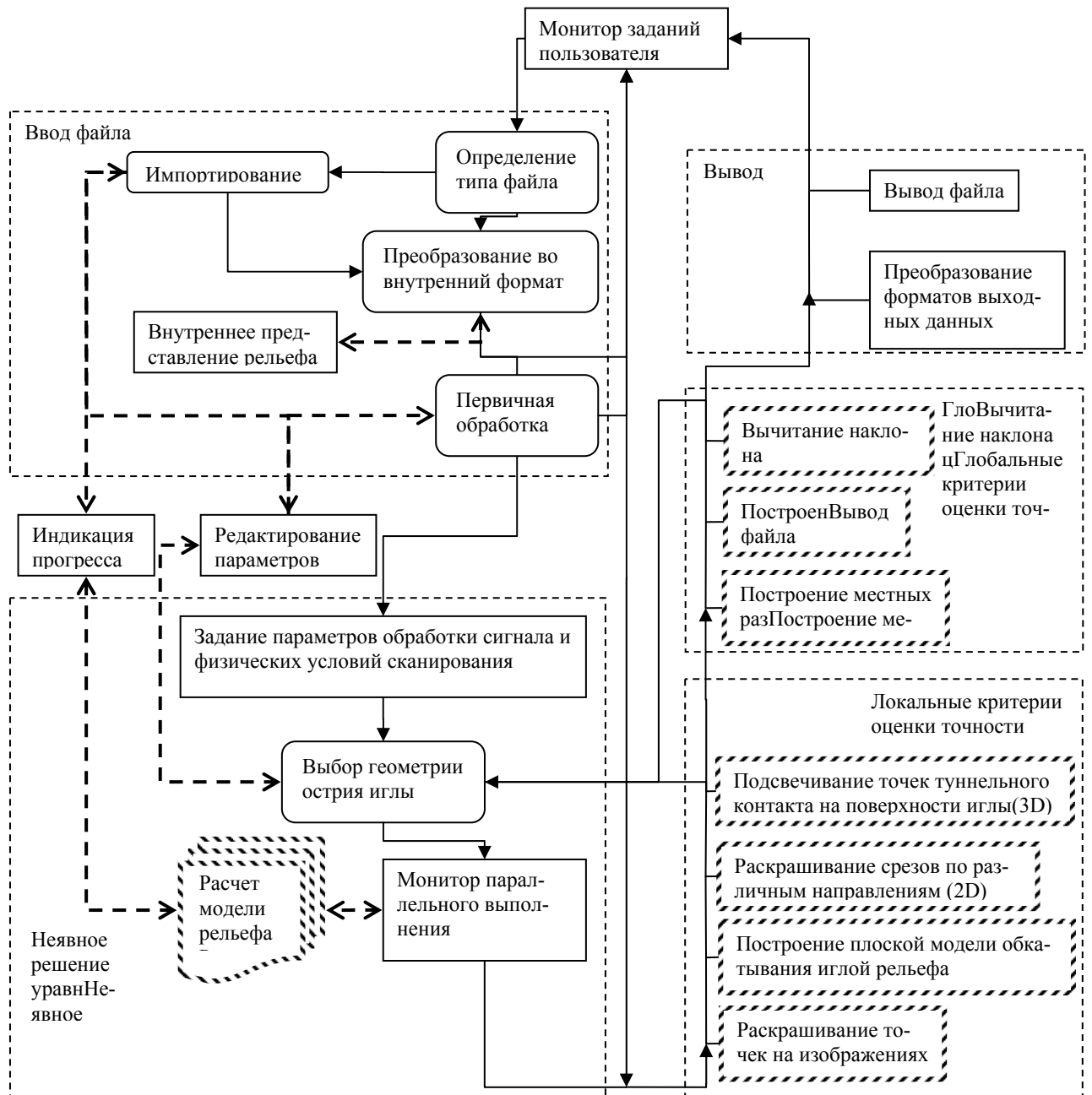
Если в дальнейшем возникает необходимость в развитии такого программного комплекса, то отсутствие ясной структуры создает значительные затруднения в работе, так как часто развитие комплекса осуществляется другими разработчиками. Решением этой проблемы является преобразование программного кода в соответствии с принципами открытых систем. Существенными факторами, сильно ограничивающим возможности применения такого подхода, являются квалификация разработчика, трудоемкость и связанные с этим затраты времени. В результате, этот вопрос часто решается в пользу создания нового комплекса.

Вариантом решения этой проблемы является частичное использование принципов открытых систем путем модификации или замены одного или нескольких простых модулей. Для этого они должны соответствовать принципам открытой архитектуры. В случае туннельной микроскопии, этот подход делает возможным использование более сложных моделей взаимодействия иглы и сканируемой подложки. Это позволяет расширить модель взаимодействия на атомно-силовую микроскопию в контактной и бесконтактной модах сканирования с тем, чтобы обеспечить определение геометрии сверхтонких острий таких кантилеверов, для которых традиционные методы калибровки неприменимы. Также делается возможным определение геометрии сверхтон-

ких острий тех кантилеверов, для которых традиционные методы калибровки неприменимы. При этом сохраняются реализованные возможности разработанного программного комплекса по неявному решению уравнения взаимодействия иглы и поверхности, которое в работе ищется путем имитации процесса измерений, по оцениванию соответствия физическим условиям сканирования, отображению и анализу поверхности.

Проект программного комплекса, который реализует эти возможности, представлен на рис. 3.3. На нем показана схема компонентов и взаимосвязь между ними в программном комплексе обработки данных зондовой микроскопии. Класс *Монитор заданий пользователя* реализуется в виде стандартного оконного меню. Он позволяет пользователю выбирать порядок обработки данных. Класс *Индикация прогресса* обеспечивает контроль пользователя за ходом обработки. Группа компонентов *Ввод файла* содержит функции определения типа считываемого файла, преобразование его во внутренний формат, и импортирования, при котором отсутствующая в графическом представлении информация дополняется пользователем. Результатом работы функции определения типа файла является экземпляр соответствующего типу файла класса-потомка базового класса. В базовом классе описываются виртуальные методы для считывания и валидации заголовка, загрузки содержимого, задания недостающих параметров и импортирования во внутренний формат. При необходимости вызывается метод импортирования с передачей ему экземпляра класса, реализующего интерфейс редактирования параметров.

На выходе функции преобразования во внутренний формат получается *совокупность высот* в точках измерения (каркас рельефа). Внутреннее представление включает в себя историю изменений моделируемого рельефа, его название и метаданные. Класс, описывающий внутреннее представление, обеспечивает реализацию события обратного вызова изменений рельефа.



Обозначения:

Имя	Класс
Действие	Функция/метод
→	Вызов
- - -	Абстрактная реализация

Функция/метод

К полученному кадру сканирования применяются фильтры первичной обработки измерений, которые уменьшают уровень черезстрочных помех. Это является необходимым, так как межстрочное осреднение туннельным микроскопом в процессе измерения не производится. В результате на изображении соседние профили рельефа иногда имеют заметные вариации по высоте. Функция первичной обработки представляет собой метод в отдельном классе. На ее вход передается каркас рельефа и желаемый тип преобразования. При необходимости, возможен также ввод параметров преобразования с помощью класса, реализующего интерфейс ввода параметров.

Группа компонентов *Неявное решение уравнения* обеспечивает обработку данных путем решения уравнения подбором геометрии острия иглы. Класс *Задание параметров обработки сигнала и физических условий сканирования* обеспечивает вызов метода объекта, реализующего этот интерфейс.

Функция *Выбора геометрии острия иглы* позволяет пользователю выбрать один из типов формы острия и указать размеры его частей. Это осуществляется путем указания потомка базового класса, описывающего виртуальные методы для доступа к результатам расчета геометрии острия. Существенным условием является то, что функции, позволяющие определить геометрию иглы и рассчитать модель рельефа, должны допускать возможность вызова в общем виде. Кроме того, возможно задание формы объекта в виде алгебраической функции, которая зависит от положения относительно системы координат.

Класс *Расчета модели рельефа* обеспечивает асинхронный вызов нескольких потоков функций моделирования рельефа. Каждая из них реализует расчет рельефа поверхности в соответствии с заданной физической моделью взаимодействия иглы и поверхности. Реализованная возможность замены класса позволяет использовать для расчета более сложную модель взаимодействия.

Расчеты по сложным физическим моделям могут занимать достаточно много времени. Для ускорения вычислений, функция расчета модели рельефа

реализована таким образом, чтобы обеспечить параллельное выполнение несколькими потоками с использованием распараллеливания процесса обработки по строкам сканирования, что, как показано в тексте диссертации, сокращает накладные расходы многопоточности.

Одной из задач разработки комплекса с заменяемыми модулями является обеспечение простой комплексной отладки нового модуля со штатными программными средствами. Разработчик комплекса может упростить комплексную отладку, если сделает доступной отладочную печать, которую он сам использовал при создании комплекса. Кроме того, важно обеспечить передачу сообщений о возникающих ошибках в заменяемых модулях во время выполнения. Эта задача решается стандартными средствами программной системы .NET.

Процесс расчета модели рельефа должен контролироваться пользователем путем реализации возможности вызова соответствующих событий. Кроме того, класс *Монитор параллельного выполнения* должен предусматривать механизм синхронизации и отрабатывать возникающие исключительные ситуации. Данный класс на вход принимает каркас рельефа, а на выходе имеет модель рельефа поверхности.

С целью обеспечения гибкости и кроссплатформенности, вывод на экран должен быть реализован в независимом модуле. Классы, предназначенные для вывода информации на экран, должны реализовывать специально предназначенный для этого интерфейс. Для реализации интерфейса удобно использовать API Win32 GDI/GDI+, GTK или OpenGL.

Группа компонентов *Локальные критерии оценки точности* представляет собой набор классов, которые обеспечивают визуализацию модели рельефа в виде вертикальных срезов, построение двумерных изображений моделей сканируемой поверхности и острия иглы, раскрашивание точек на изображениях с учетом выбранных критериев оценки соответствия результатов моделирования физическим условиям. Задачей программных модулей, входящих в рассматриваемую группу является описание места и функций инте-

рактивного интерфейса пользователя и средств визуализации получаемых результатов для обеспечения возможностей оценки им степени соответствия геометрии иглы и выбранных критериев физических условий сканирования получаемому изображению нанорельефа.

Группа компонентов *Глобальные критерии оценки точности* объединяет классы, с помощью которых выполняется построение двумерного и трехмерного изображений рельефа и исследования его геометрии с помощью местных видов и разрезов. Построение изображений рельефа позволяет применять для оценки соответствия принятой модели рельефа глобальным критериям такие параметры, которые определяются формой и размерами наблюдаемых геометрических объектов. Также оно позволяет исследовать геометрию наноборозований на отсканированной поверхности.

Группа компонентов *Вывод* отвечает за запись обработанных данных в файл в различных форматах. При работе с этими компонентами пользователю необходимо учитывать изложенные выше проблемы форматов файлов и выбирать такой формат, который мог бы сохранить точные величины высот, а также метainформацию об условиях сканирования и о файле.

§4. Выбор языка программирования для реализации комплекса

Построение заменяемого модуля и его программного окружения на основе объектно-ориентированного подхода позволяет использовать поддержку замены некоторыми языками программирования высокого уровня (например, C#, Python, Java). Удобство реализации этой возможности в языке C# заключается в автоматическом формировании требований к структуре заменяемого модуля (они хранятся в метainформации файла сборки) и последующей проверке соответствия нового модуля этим требованиям на этапе компиляции. Кроме того, платформа .NET, обеспечивающая выполнение программ, написанных на языке C#, позволяет проверять корректность параметров, передаваемых функциям, на этапе исполнения во время отладки (ме-

ханизм Code Contract, подразумевающий формулирование ограничений на параметры функций непосредственно в коде и их автоматическую проверку средой исполнения в процессе работы программы [48]).

Дополнительные трудозатраты по преобразованию заменяемого модуля и его окружения к объектно-ориентированному виду следует признать целесообразными, так как это упростит задачу разработчику нового модуля.

Заменяемость модуля вполне может быть обеспечена средствами языка C++ [49]. Однако, использование только языка C++ нецелесообразно, так как приводит к необходимости решать достаточно сложные задачи разработки программного кода. К ним относятся освобождение памяти, которое в C++ выполняется вручную, и необходимость универсализации программного кода для различных операционных систем, в том числе для разных версий ОС Windows. Грамотное решение этих задач предполагает определенную квалификацию разработчика. Отсутствие грамотного решения этих проблем существенно ограничивает возможности применения программного комплекса с большим объемом программного кода и затрудняет или ограничивает его дальнейшее развитие.

По этой причине, при разработке развитых программных комплексов целесообразно воспользоваться средствами языка C#, которые реализуются в программной среде .NET. Кроме этого, средства языка упрощают программисту использование следующих возможностей: реализация заменяемого модуля, средства разработки интерфейса с пользователем, поддержка многоуровневых файлов конфигурации [46], локализация, средства отладки, реализация многопоточности. Трудозатраты на освоение языка C# в среде .NET намного ниже, чем время, затраченное на решение упомянутых задач средствами языка C++. В табл. 3.2 сравниваются возможности языков C, C++ и C# в контексте перечисленных средств разработки.

Табл. 3.2. Сравнение возможностей языков C, C++ и C#.

	C	C++	C#
Спецификация модуля	Заголовочный файл, описывающий структуру		Спецификация класса хранится в метаданных сборки
	модуля	класса	
Работа с памятью	malloc / free	new / delete	new; сборщик мусора
Совместимость с различными ОС	Требуется перекомпиляция (иногда также адаптация)		Требуется поддержка среды выполнения
Проверка корректности аргументов в функциях	Проверка условий вручную, а также		Проверка условий с помощью Code Contract [46] или вручную; генерация исключений
	выдача кода возврата	генерация исключений	
Динамическая компиляция	Обеспечивается сторонними библиотеками		Поддерживается средой выполнения
Хранение настроек программы	Обеспечивается ОС или сторонними библиотеками		Обеспечивается средой выполнения (класс ConfigurationManager)
Многопоточность и синхронизация	Обеспечивается API операционной системы		Платформенно-независимые средства среды выполнения; синтаксическая конструкция lock

Из приведенной таблицы следует, что преимуществом написания программ и компонентов на языке C#, является полная поддержка описанных

возможностей платформой, обеспечивающей их выполнение, и не требует вмешательства в среду.

§5. Модифицируемость программного кода

Предусмотренная разработчиком возможность изменять написанный программный код существенно повышает ценность комплексов, предназначенных для исследовательских целей. Возможности современных средств разработки позволяют существенно упростить процесс создания программ с модифицируемым кодом.

Можно условно выделить три уровня, которые обеспечивают изменение исполняемого кода: уровень замены констант, определяющих логику работы, уровень замены формул и уровень замены модуля. Каждый последующий уровень требует повышения трудоемкости реализации наряду с расширением возможностей.

5.1. Уровень замены констант

.NET Framework поддерживает возможность хранения настроек и констант в файле конфигурации. Это XML-документ, содержащий системную информацию об окружении, а также пользовательские разделы [46]. Настройкам соответствует адрес XPath

`/configuration/appSettings/add,`

причем в элементе `add` атрибут `key` означает название настройки, а `value` – ее значение. Считывание настроек осуществляется на языке C# с помощью свойства

`System.Configuration.ConfigurationManager.AppSettings,`
возвращающего ассоциативный массив пар «название настройки – значение». Пример заменяемых констант показан ниже.

В этом примере показаны константы программного комплекса, которые описывают уровень допустимых отклонений от физических условий сканирования (строка 10) и цвет точек рельефа, в которых условия соблюдены

(строка 11 задает зеленый цвет таких точек), а также интервал, задающий

```

1  <?xml version="1.0" encoding="utf-8" ?>
2  <configuration>
3      <appSettings>
4          <add key="subtractPlane" value="y+z-1 = 0"/>
5          <add key="zDisplayRange_nm" value="[0;10]" />
6          <add key="primaryRelief_Color" value="White" />
7          <add key="secondaryRelief_Color" value="Yellow"/>
8          <add key="mark1nm_Color" value="Green" />
9          <add key="lowestPoint_Color" value="Magenta" />
10         <add key="borderDelta2_nm" value="0.30" />
11         <add key="border2_Color" value="Green" />
12     </appSettings>
13 </configuration>

```

подмножество выводимого рельефа, соответствующего диапазону высот (строка 5). Кроме того, строка 4 задает уравнение плоскости, соответствующей наклону рельефа по умолчанию; строки 6, 7, 8 и 9 определяют правила раскраски в органе управления, отвечающем за рисование профилей рельефа поверхности образца.

В программном комплексе окно изменения файла конфигурации вызывается с помощью меню. Оно открывается в виде отдельной вкладки, в которой показано содержимое конфигурационного файла и имеются органы управления, позволяющие сохранить или отменить изменения (см. гл. 4).

5.2. Уровень замены формул

В платформе .NET Framework имеется возможность расширить функциональность программного кода кодом, указанным пользователем. Это позволяет модифицировать модуль расчетов путем указания алгебраической формулы или выражения с логическими условиями без компиляции пользователем. В языке C# эта возможность называется динамической компиляцией и реализуется следующим образом:

```
// Описание базового класса
public class BaseClass
{
    public BaseClass() {}
}

. . . . .

CSharpCodeProvider codeCompiler = new
CSharpCodeProvider();

// Добавление ссылок на сборки
CompilerParameters parameters = new
CompilerParameters();

// Добавление ссылки на текущую сборку
string path
=Assembly.GetExecutingAssembly().Location;

parameters.ReferencedAssemblies.Add(path);

// Компиляция
CompilerResults result
=codeCompiler.CompileAssemblyFromSource(parameters,
code);

// Создание экземпляра класса
string className = "Macro"

object objMacro
=result.CompiledAssembly.CreateInstance(className);
```

Здесь `BaseClass` – класс, который может быть наследован в компилируемом коде. `CSharpCodeProvider` – класс, в котором реализованы средства динамической компиляции. Методу `CompileAssemblyFromSource`, осуществляющему компиляцию, передаются её параметры и `code` – параметр, содержащий программный код для компиляции. Переменная `className` содержит название создаваемого класса. Пример кода, описывающего такой класс, показан ниже.

```
using System;

using CustomAssembly;
// Пространство имен добавленной сборки (см. выше)

namespace MacroNamespace
{
    public class Macro : BaseClass
    {
        public void Run()
        {
            Console.WriteLine("Hello, world!");
        }
    }
}
```

Пользователю достаточно указать только тело метода в заменяемом классе (в указанном примере это строка кода в методе `Run`), которое в таком случае будет встроено программным комплексом в шаблон содержания кода. В указанном примере это весь код, кроме строки, описывающей метод `Run`.

5.3. Уровень замены модуля

Наиболее простым способом организации замены модуля является выделение его программного кода в динамическую библиотеку, локализованную в отдельном файле. Этот способ не обязывает строить и модуль, и программный комплекс в целом, на принципах объектно-ориентированного программирования. Достаточно обеспечить соответствие спецификаций заменяемого модуля структуре данных всего комплекса и правилам вызова функций заменяемого модуля.

Для скомпилированной программы на языке C# можно явно указать заменяемый модуль в файле конфигурации:

```
<configuration>
<runtime>
  <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
    <dependentAssembly>
      <assemblyIdentity name="module"/>
      <codeBase href="module2.dll"/>
    </dependentAssembly>
  </assemblyBinding>
</runtime>
</configuration>
```

Здесь `module` – название сборки, которую необходимо заменить. Строка «`module2.dll`» – URI (Uniform Resource Identifier, унифицированный идентификатор ресурсов) файла со сборкой, которая заменяет сборку `module`. Существенным требованием к заменяемому файлу является полное совпадение имени сборки и её пространства имён с исходной сборкой. Таким образом, в приведенном примере файл `module2.dll` содержит сборку с именем «`module`».

Кроме описанного способа, можно также осуществить замещение файла программного комплекса пользовательским файлом. В системе .NET на

замену подписанного модуля, имеющего собственный открытый ключ, налагаются некоторые ограничения [46].

§6. Проектирование многопоточности

Использование построчной обработки данных в рассматриваемой задаче делает возможным эффективно реализовать многопоточность, распределив задачи для строк по различным ядрам/процессорам. Следует заметить, что существующие средства обработки результатов измерений зондовой микроскопии многопоточную обработку данных не поддерживают.

В языке C# имеется удобный аппарат, с помощью которого можно осуществлять многопоточную обработку. Стандартным образом она осуществляется двумя способами:

1. Реализацией модели асинхронного программирования с помощью статических методов класса `System.Threading.Tasks.Future<T>` и реализации интерфейса `System.IAsyncResult`.

Шаблонный класс `System.Threading.Tasks.Future` из библиотеки параллельного программирования `Parallel Extensions` для `.NET Framework` наследует класс `System.Threading.Tasks.Task`. Он отличается тем, что представляет собой вычисления, возвращающие значение (в отличие от класса `Task`, который не возвращает значений, как и класс `Thread`). При обращении к `Future<T>` за возвращаемым значением, оно либо сразу будет возвращено (если оно уже вычислено задачей к этому моменту), либо произойдет блокирование работы вызывающей стороны до тех пор, пока требуемое значение не будет вычислено [46]. Чтобы узнать, вычислено ли значение задачей `Future<T>`, можно обратиться к его свойству `IsCompleted`. `Future<T>` может работать с анонимными методами. В таком случае происходит создание объекта класса `Future<T>` без явного указания тела задачи. При этом возвращаемое значение такой задачи становится готовым при присваивании значения свойству `Value` этой задачи. Если результат выполнения `Future<T>` ожидали некоторые дру-

гие задачи, они будут разблокированы этим присваиванием. Необходимо отметить, что поле Value объекта Future<T> допускает только однократное присваивание. Если же по каким-то причинам нет возможности вычислить значение, возвращаемое задачей Future<T>, то сообщение об этом может быть передано через поле Exception данной задачи.

Рассмотрим пример работы с классом Future<T>. Достаточно часто в приложениях, обрабатывающих какие-либо данные, можно встретить следующий сценарий работы: приложение получает набор данных для обработки, обрабатывает его, затем выполняет какую-то дополнительную работу и лишь после этого использует результаты обработки данных. В этом случае, Future<T> позволит значительно ускорить выполнение такого сценария - во-первых, за счет параллельной обработки данных, а во-вторых, за счет параллельного выполнения дополнительной работы и непосредственно обработки:

```
// Задание массива результатов обработки
var data = new Future<int> [10000];
// Запуск их параллельной обработки
Parallel.For(0, data.Length, i =>
{
    data[i] = Future.Create(() => Compute(i));
});
// Продолжение работы программного комплекса
```

В этом примере вычисления проводятся с помощью метода Compute() .

2. Реализацией многопоточности явным образом вручную путем создания потоков командой new Thread(...) и запуском их на выполнение методом Start().

Описанные подходы реализации многопоточности имеют много общего, и в описываемом в диссертации программном комплексе используется решение, в котором реализуются механизмы, описанные в обоих способах. В це-

лом же эти приемы целесообразно использовать для вычислений, требующих много процессорных ресурсов, а также для избегания блокировки пользовательского интерфейса.

При построчной обработке данных возникает проблема синхронизации работы ядер, так как при одновременном доступе к общим (разделяемым) ресурсам требуется обеспечение эксклюзивного доступа к ним с целью устранения возможного состояния гонки [50]. К таким ресурсам относятся массив, описывающий модель рельефа, а также пользовательский интерфейс (в части органов управления, ответственных за управление процессом вычисления). Кроме того, требуется система оповещения выполняющихся потоков об исключительных ситуациях, которые могли возникнуть в одном из них (ошибка или событие, связанное с пользовательским интерфейсом). Синхронизация вычислительного процесса, выполняющегося в разных ядрах и потоках, осуществляется с помощью стандартного механизма семафоров и событий, который описан в [46].

Предлагаемое решение по реализации многопоточности может применяться в тех случаях, когда вычислительная задача обработки данных для потока не зависит от результатов расчетов для других потоков. Этим свойством обладают те задачи, в которых решение для каждой точки может быть получено применением функционала только к исходным данным.

Рассматриваемая в работе задача обладает этим свойством, так как точка поверхности модели рельефа выбирается на поверхности той туннельной оболочки, для которой достигается минимум вертикальной координаты по всем туннельным оболочкам, которые построены в соседних точках измерения:

$$\bar{z}_i = \min_{j: \rho(x_i, y_i, x_j, y_j) < R} S(x_j, y_j, z_j, G),$$

где $\bar{z}_i$ – вертикальная координата точки моделируемого рельефа, горизонтальные координаты которой совпадают с узлом сетки измерений (x_i, y_i) ; (x_j, y_j, z_j) – измеренные координаты рельефа (за которые принимается траектория

движения иглы); $\rho(x_i, y_i, x_j, y_j)$ – расстояние между узлом сетки измерений (x_i, y_i) , в которой вычисляется модель рельефа и соседними узлами (x_j, y_j) ; R – радиус туннельной оболочки; $S(x_j, y_j, z_j, G)$ – поверхность туннельной оболочки в точке траектории движения иглы (x_j, y_j, z_j) и геометрии острия G .

Выводы третьей главы

1. Обработка данных сканирования, связанная с неявным решением уравнения взаимодействия иглы с поверхностью, содержит несколько потоков интерактивного выбора, связанных с подбором решения и оценкой результатов моделирования. При этом могут использоваться критерии поточечного сравнения с параметрами процесса измерения, так и глобальные критерии, которые учитывают выполнимость геометрических соотношений и физических законов.
2. Точность получаемого решения зависит от подробности геометрических и физических моделей, используемых на этапе выбора решения, вычисления решения и его оценки. Каждая из этих задач может быть реализована с помощью различных по сложности программных решений: заменой констант, модификацией расчетных формул и полной замены вычислительной модели.
3. Использование сложных геометрических и физических моделей обуславливает увеличение производительности вычислений. Эффективность использования нескольких ядер/процессоров в расчетах основывается на учете особенностей структуры данных измерений.
4. Язык C# позволяет существенно уменьшить трудоемкость разработки развитых программных комплексов зондовой микроскопии, которые предусматривают модификацию программного кода и реализацию многопоточности. Это достигается за счет поддержки тех функций, которые в предыдущих языках выполнялись вручную.

Глава 4. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ КОМПЛЕКСА ДЛЯ ОБРАБОТКИ ДАННЫХ СКАНИРОВАНИЯ

В главе описывается опытная программная реализация модели комплекса обработки данных туннельной микроскопии, средства пользователя для настройки процесса обработки данных и приводятся результаты применения этого программного комплекса в научных исследованиях. Обсуждаются возможности комплекса по очищению изображения от шумов, артефактов, более точному воспроизведению формы и размеров нанообразований и структуры их расположения на исследуемой поверхности.

Особенность решаемой задачи на этапе поиска оптимального сочетания «игла-рельеф поверхности» состоит в том, что оценку качества пользователь производит самостоятельно. В связи с этим, в программном комплексе реализованы специальные средства, которые предоставляют возможность объективной оценки и сравнения различных вариантов решений. Умелое использование этого инструментария ускоряет нахождение оптимального решения.

§1. Внутренний формат данных

Информация об измеренном рельефе представляется в виде массива высот, упорядоченного по номерам профилей рельефа. Внутренний формат данных также содержит раздел с метаинформацией о рельефе поверхности и параметрах сигнала. Пользователь имеет возможность сохранять и считывать данные во внутреннем формате. Структура внутреннего формата указана в таблице 4.1. Файлы, содержащие данные во внутреннем формате, имеют расширение HGT.

Таблица 4.1. Внутренний формат представления данных.
--

Таблица 4.1. Внутренний формат представления данных.

const int HeaderPart1Size = 0x18	Данная константа задает размер первой части заголовка
Int64 Signature1	Идентификатор формата файла (всегда соответствует ASCII-представлению 8 символов 'm', 'f', 'd', 0x20, 't', 'u', 'n', 'n')
Int64 Signature2	Идентификатор версии и формата файла (всегда соответствует ASCII-представлению 8 символов 'e', 'l', 0x20, 'n', 'a', 'n', 'o', '.')
Int16 zero	Данное значение служит для обозначения конца строки, указанной выше, и для выравнивания по границе количества байт в заголовке предыдущей версии (версии 2)
Int16 eof	Еще один нулевой байт для выравнивания и символ «конец файла» в кодировке ASCII (0x26)
Int32 version	Идентификатор версии файла (0x00000003)
byte[] headerPart2	Вторая часть заголовка: сериализованная информация о рельефе (экземпляр класса ReliefParameters)
byte[] relief	Сериализованная информация о рельефе (экземпляр класса Relief)

На рис. 4.1 показаны данные, используемые в программном комплексе. Класс `WorkspaceParameters` хранит информацию о сеансе работы. В переменной `heights` класса `ReliefHistoryLayers` находится история всех изменений, проводимых с рельефом и его моделью. Переменная `config` класса `mfd.Library.Probes.Parameters.Parameter` содержит информацию о последней модели иглы, принимавшей участие в обработке рельефа с помощью описанного в гл. 3 алгоритма. Переменные `minimap`, `navigation`, `draw`, `feedback` отвечают за рисование рельефа поверхности в окне программного комплекса.

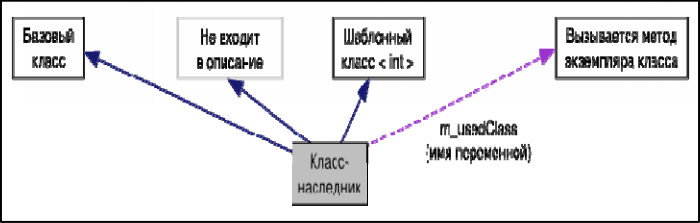


Рис. 4.1. Основные данные, схема вызовов и наследования в программном комплексе.

§2. Органы управления и пользовательский интерфейс

Программный комплекс предназначен для интерпретации измерений, выполненных туннельным микроскопом, и представленных в одном из описанных ранее форматов – NWF, BMP или внутреннем HGT.

Считывание файла с данными осуществляется с помощью пункта меню главного окна *File*→*Open*. В открывшемся окне нужно выбрать расширение и указать имя файла, найдя его в списке (рис. 4.6). Если загружаемый файл имеет формат NWF или HGT, то все внутренние параметры загружаются автоматически.

Если исходный файл закодирован в формате BMP, то его загрузка происходит путем диалога. Сначала требуется загрузить файл с палитрой, а потом в появившемся модальном диалоге задать два измерения изображения в нанометрах: длину (*length_nm* на рис. 4.7) или ширину изображения (*width_nm*), а также максимальную высоту (*height_nm*). Эти величины используются для масштабирования диапазона изменений высоты рельефа поверхности.

Программный комплекс обладает оконным интерфейсом пользователя, и его главное окно имеет вид, показанный на рис. 4.2. Управление обработкой информации осуществляется с помощью меню. Отображение текущих данных о рельефе поверхности происходит во вкладке *Display*. Предусмотрена возможность отображения профилей рельефа, двумерного и трехмерного изображений рельефа. Переключение режима отображения осуществляется через подпункты меню *View*→*Display Mode*. Ниже описаны основные возможности работы с главным окном.

На рис. 4.2 показан состав меню, органы управления и окна ввода параметров, относящихся к обработке и визуализации данных. Функции меню и его структура соответствуют сценарию работы с главным окном. Состав и логика работы органов управления выбраны таким образом, чтобы пользователь мог контролировать процесс решения задачи определения геометрии

острия иглы и эффективно управлять параметрами этого процесса в интерактивном режиме.

С правой стороны главного окна отображается двумерное изображение рельефа. На данном изображении высота рельефа кодируется с помощью выбранной палитры. Пользователь может получить изображение профиля рельефа по произвольно указанному направлению. Направление среза указывается им на двумерном изображении рельефа.

Предусмотрена возможность рисования линий уровня на двумерном изображении рельефа. Для этого требуется указать относительное положение линии уровня минимальной и максимальной высоты рельефа в процентах (рис. 4.2, п.10).

Исходный профиль рельефа и профиль, полученный в результате обработки (рис. 4.12) выводятся на экран вместе. На изображение профиля можно выводить высоту выбранного узла в нанометрах (рис. 4.2, п.6).

На рис. 4.2, п. 4 показаны органы управления для задания параметров геометрической модели туннельной оболочки пользователем. Выбор определенной геометрической модели иглы (и туннельной оболочки) осуществляется через меню *Filtering* → *Restoration* → *Using Probe Simulation* (рис.4.2, п.1). Результатом работы модуля 2 программной модели является геометрическая модель туннельной оболочки (см. рис. 4.3).

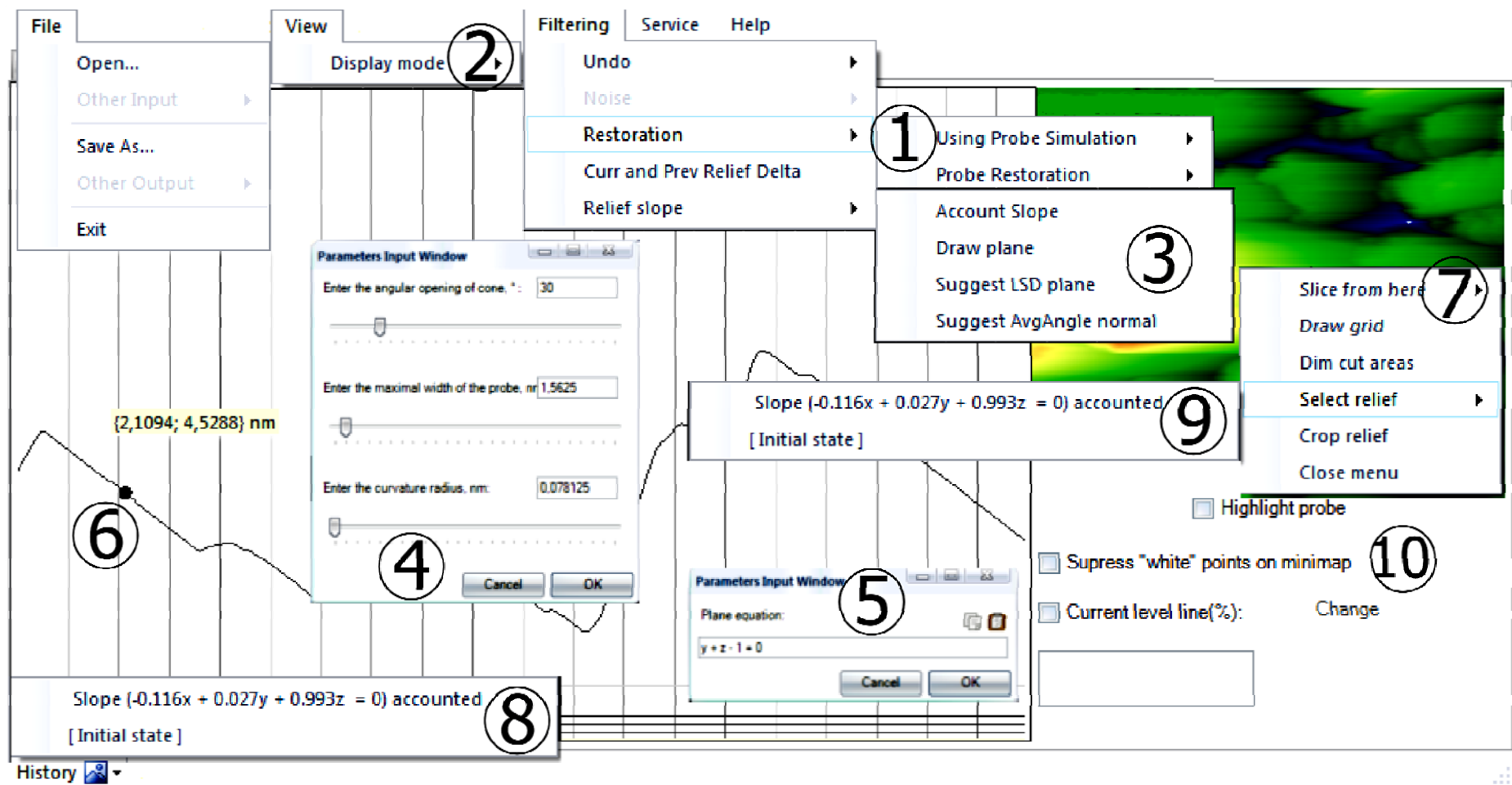


Рис. 4.2. Окно программы в режиме работы с профилями рельефа.

§3. Задание туннельной оболочки острия иглы

Условия взаимодействия зонда и поверхности, при которых выполнялось сканирование поверхности, учитываются путем задания величины туннельного зазора. Его величина должна соответствовать тем параметрам сканирования, которые записаны в параметрах произведенных измерений. Величина туннельного зазора полагается постоянной.

С помощью пункта меню *Filtering*→*Tunnel gap* пользователь может задать величину туннельного зазора. В таком случае эта величина добавляется к радиусу закругления иглы. Либо пользователь может с помощью данного меню указать величину туннельного зазора как 0, после чего самостоятельно работать с моделью туннельной оболочки. Таким образом, туннельный зазор можно также учитывать отдельно, задавая туннельную оболочку самостоятельно. Если в процессе задания её параметров указываемое пользователем значение величины меньше туннельного зазора, программный комплекс выдает предупреждение.

После выбора пользователем величины туннельного зазора, программным комплексом формируется модель туннельной оболочки по геометрии острия иглы в соответствии с описанной выше логикой. Программный комплекс предоставляет пользователю возможность выбора модели острия иглы. Реализованы следующие модели:

1. Цилиндр (меню *Filtering*→*Restoration*→*Using Probe Simulation*→*Spherical*)

Эта модель иглы является трехмерной и однопараметрической. Параметром является радиус цилиндра. Игла представляет собой цилиндр с полусферой на конце и описывается соотношением

$$\begin{cases} z = z_0 + r - \sqrt{r^2 - (x - x_0)^2 - (y - y_0)^2}, & (x - x_0)^2 + (y - y_0)^2 < r^2 \\ z \geq z_0 + r, & (x - x_0)^2 + (y - y_0)^2 = r^2 \end{cases},$$

где r – радиус цилиндра, а $\{x_0, y_0, z_0\}$ – координаты острия иглы (рис. 4.3, а).

2. Двумерная парабола (меню *Filtering→Restoration→Using Probe Simulation→Parabolic (2D)*)

Параболическая модель иглы является первым приближением реальной иглы (рис. 4.3, б). Модель имеет один параметр и описывается формулой

$$(y-y_0)^2 = 2p(x-x_0),$$

где p – параметр, а $\{x_0, y_0\}$ – координаты острия иглы на очередном профиле рельефа.

Параболическая игла обладает рядом недостатков. Во-первых, ветви параболы имеют особенность сильно расходиться с ростом координаты. Поэтому вне ограниченной области вблизи острия модель теряет адекватность. Во-вторых, ширина реальной иглы, начиная с некоторого уровня, стабилизируется (рис. 3.9), чему также противоречит данная модель иглы.

Настройка данной модели затруднена. Например, невозможно настраивать уровень расхождения ветвей.

3. Двумерный цилиндр (меню *Filtering→Restoration→Using Probe Simulation→Cylinder-Circle (2D)*)

Модель иглы в форме двумерного цилиндра с полуокружностью на конце описывается соотношением

$$\begin{cases} y = y_0 + r - \sqrt{r^2 - (x - x_0)^2}, & |x - x_0| < r \\ y \geq y_0 + r, & |x - x_0| = r \end{cases},$$

где r – радиус цилиндра, а $\{x_0, y_0\}$ – координаты острия иглы на очередном профиле рельефа (рис. 4.3, в).

4. Конус вращения, не ограниченный по максимальной ширине (меню *Filtering→Restoration→Using Probe Simulation→Cone-sphere*)

Указанная модель представляет собой усеченный конус со сферой на острие. Для задания имеется два параметра – угол раствора конуса α и радиус сферы r . Модель описывается следующей системой:

$$\begin{cases} z = z_0 + r - \rho_{2D}, & \rho_{2D} \leq r \\ z = z_0 - r \left(\frac{1}{\sin(\alpha/2)} - 1 \right) + \rho_{2D} \operatorname{ctg}(\alpha/2), & r < \rho_{2D} \end{cases},$$

где r – радиус сферической части иглы, α – угол раствора конической части иглы, $\{x_0, y_0, z_0\}$ – координаты острия иглы, а ρ_{2D} – расстояние от интересующей точки до оси симметрии иглы:

$$\rho_{2D} = \sqrt{r^2 - (x - x_0)^2 - (y - y_0)^2}.$$

5. Конус вращения (меню *Filtering→Restoration→Using Probe Simulation→Cylinder-cone-sphere*)

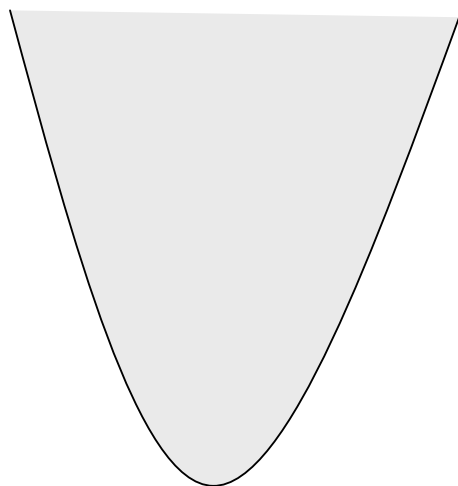
Данная модель является трехмерной и трехпараметрической. Задаются три параметра: диаметр иглы, диаметр сферической части и угол раствора конуса. Наличие нескольких параметров позволяет получать более точное решение, чем в случае цилиндрической иглы. Соотношение, описывающее иглу, имеет вид

$$\begin{cases} z = z_0 + r - \rho_{2D}, & \rho_{2D} \leq r \cos(\alpha/2) \\ z = z_0 - r \left(\frac{1}{\sin(\alpha/2)} - 1 \right) + \rho_{2D} \operatorname{ctg}(\alpha/2), & r \cos(\alpha/2) < \rho_{2D} < w/2, \\ z \geq z_0 - r \left(\frac{1}{\sin(\alpha/2)} - 1 \right) + \frac{w}{2} \operatorname{ctg}(\alpha/2), & \rho_{2D} \geq w/2 \end{cases}$$

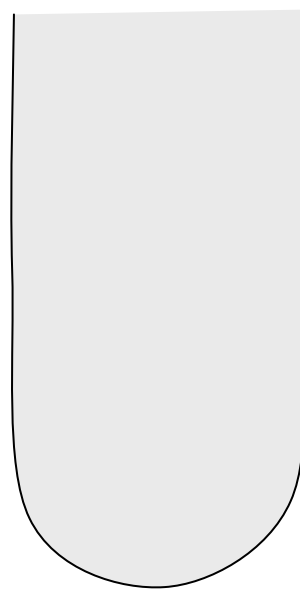
где r – радиус сферической части иглы, α – угол раствора конической части иглы, w – максимальный диаметр конической части, $\{x_0, y_0, z_0\}$ – координаты острия иглы (рис. 4.3, г), а ρ_{2D} определяется так же, как и в предыдущей формуле.

Соединение сферической и конической частей иглы происходит по условию соблюдения равенства производной в точках перехода сферической части в коническую.

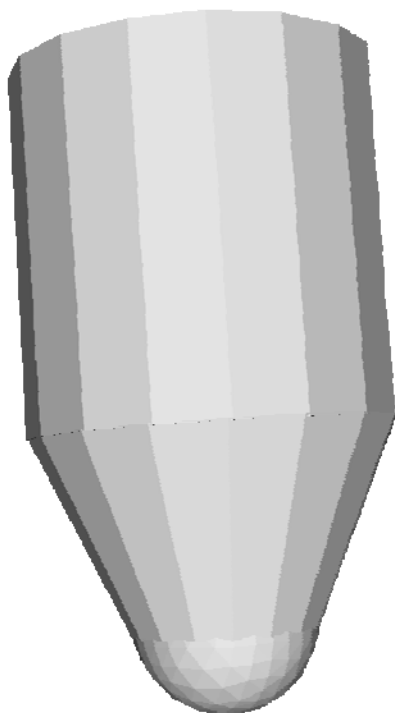
Кроме указанных выше, в программном комплексе предусмотрена возможность задания произвольной геометрической модели острия иглы в аналитическом виде путем задания функции вида $z = f(x_0, y_0, z_0; x, y)$.



а.



б.



в.



г.

Рис. 4.3. Модели острий игл.

§4. Средства оценивания точности определения рельефа

В качестве критерия для оценки точности модели острия иглы используется система соотношений (2.4). Для обеспечения наглядности представления процесса подстройки туннельной оболочки, в программном комплексе реализован алгоритм подсветки величины отклонения от них. Для каждой точки исходного рельефа ищется расстояние от туннельной оболочки, построенной в данной точке, до моделированного рельефа поверхности. Цвет назначается в зависимости от полученного расстояния.

В программном комплексе описанный алгоритм реализован в многопоточном виде. Реализация параллельного выполнения в данном случае в точности соответствует архитектуре многопоточности алгоритма построения модели поверхности.

В качестве геометрии острия иглы может быть выбрана та модель, для которой система равенств (2.4) нарушается в наименьшем числе контрольных точек. Если она нарушается для слишком большого количества контрольных точек, то выбранную модель нельзя считать удовлетворительной, и она требует уточнения.

В модуле реализован механизм интерактивного контроля точности получаемого решения. Он включает в себя построение графика разности значений высоты исходного и моделированного профилей рельефа; просмотр смоделированного положения иглы при сканировании; настройка подсветки линий уровня и нанометровой сетки; подавление пикселей, отвечающих за помеченные точки рельефа, на его двумерном изображении (рис. 4.2, п.10). Это делает наглядными изменения рельефа до и после обработки.

Разработанные графические средства позволяют визуализировать отклонения между разными моделями поверхностей, полученными с помощью разных туннельных оболочек. Указанную возможность удобно использовать для оценки точности представления получаемых результатов.

Кроме этого можно вывести исходный рельеф с обработанным в одном окне. Моделируется также исходное положение иглы и ее движение при сканировании.

§5. Организация параллельных вычислений

Алгоритм построения модели рельефа поверхности, которая задается соотношением (2.5), основан на построении огибающей плоскости ко всем положениям, которые игла занимала при сканировании. Для каждой точки рельефа ищется туннельная оболочка, имеющая наименьшее значение по вертикальной координате. В результате получается кусочно-гладкая поверхность, составленная из частей туннельных оболочек, являющаяся моделью измеренной поверхности для выбранной модели формы острия иглы.

Рассмотренный алгоритм допускает параллельное выполнение на уровне ядер и процессоров. Многопоточность достигается назначением потока с номером $i \bmod \text{количество_потоков}$ на вычисление значения для i -й линии каркаса рельефа. Количество потоков, используемых для вычислений, задано соотношением

$$\text{количество_потоков} = \max(\text{количество_процессоров} \cdot \text{количество_ядер}, 1),$$

причем количество ядер соответствующего процессора умножается на 2 при наличии у него функции Intel HyperThreading. Синхронизация осуществляется использованием стандартного механизма эксклюзивной блокировки System.Threading.Monitor. Для этого используется синтаксическая конструкция **lock() { }**, примененная к специально создаваемому для этого объекту блокировки. Ожидание окончания выполнения потоков осуществляется с использованием семафора, в котором начальному значению соответствует количество потоков, а каждый поток, закончивший выполнение, уменьшает значение на 1. Сингализирование об исключительных ситуациях реализовано с использованием механизма ручных событий (System.Threading.EventWaitHandle). Использование указанных механизмов rea-

лизовано стандартным образом. Ниже приведен пример программного кода, который используется в процессе многопоточной обработки.

```
try {
    simulationCompleted = new Semaphore(0, threads-1);
    for (j = 0; j < p.R.width_px; j++)
        { Вычисление параметров иглы на рельефе в данной точке }
    lock (result)
        { Запись значения в новый объект result класса Relief }
    if (cancel.WaitOne(TimeSpan.Zero))
        throw new OperationCanceledException();
}
catch (Exception) {
    cancel.Set();
}
simulationCompleted.Release();
```

Целесообразность реализации режима многопоточной обработки подтверждается уменьшением затрат времени на построение модели поверхности практически пропорционально числу задействованных процессоров. В результате параллельной обработки данных на 4-х ядерном процессоре, время вычисления в параллельных алгоритмах описываемого программного комплекса уменьшается в несколько раз. Например, время решения задачи восстановления рельефа поверхности для часто встречающихся случаев описанным выше ресурсозатратным алгоритмом удалось снизить с примерно 10 мин при выполнении в одном потоке до 2-3 минут при выполнении в 4-х потоках на данном процессоре, что соответствует ускорению около четырех раз. Для типичного случая сокращение времени примерно пропорционально количеству параллельно выполняющихся потоков (см. табл. ниже).

Таблица 4.2. Время вычисления на разных процессорах и разном количестве потоков.

Процессор, Модель острия	тип	Intel Pentium 4 1500 Mhz	AMD Phenom 9550 (Mhz), 1 поток	AMD Phenom 9550 (Mhz), 4 потока
Коническая, $r = 0,5$ нм, $w = 2$ нм, $\alpha = 60^\circ$		305 сек	240 сек	64 сек
Коническая, $r = 1,3$ нм, $w = 4$ нм, $\alpha = 60^\circ$		1708 сек	658 сек	177 сек
Двумерная цилиндриче- ская, $r = 1,3$ нм		86 сек	39 сек	10 сек

На рис. 4.4 показаны классы и взаимосвязи между ними при организации многопоточности. Класс `Async` отвечает за инициализацию вычислений в очередном потоке. Переменная `debug` предназначена для индикации прогресса и обеспечивает возможность отмены выполнения вычислений. В переменной `relief` хранится массив высот исходного рельефа поверхности, в `simulatedRelief` — модель рельефа. Переменные `sourcePointContribution`, `isLowestPoint`, `probesHavingConeInResult` и `whitePointDistance` отвечают за хранение различной информации о выполнении условий сканирования. Эта информация заносится в переменные при построении модели рельефа поверхности и позже используется при отображении профилей рельефа и его двумерного изображения, раскрашенного по высоте.

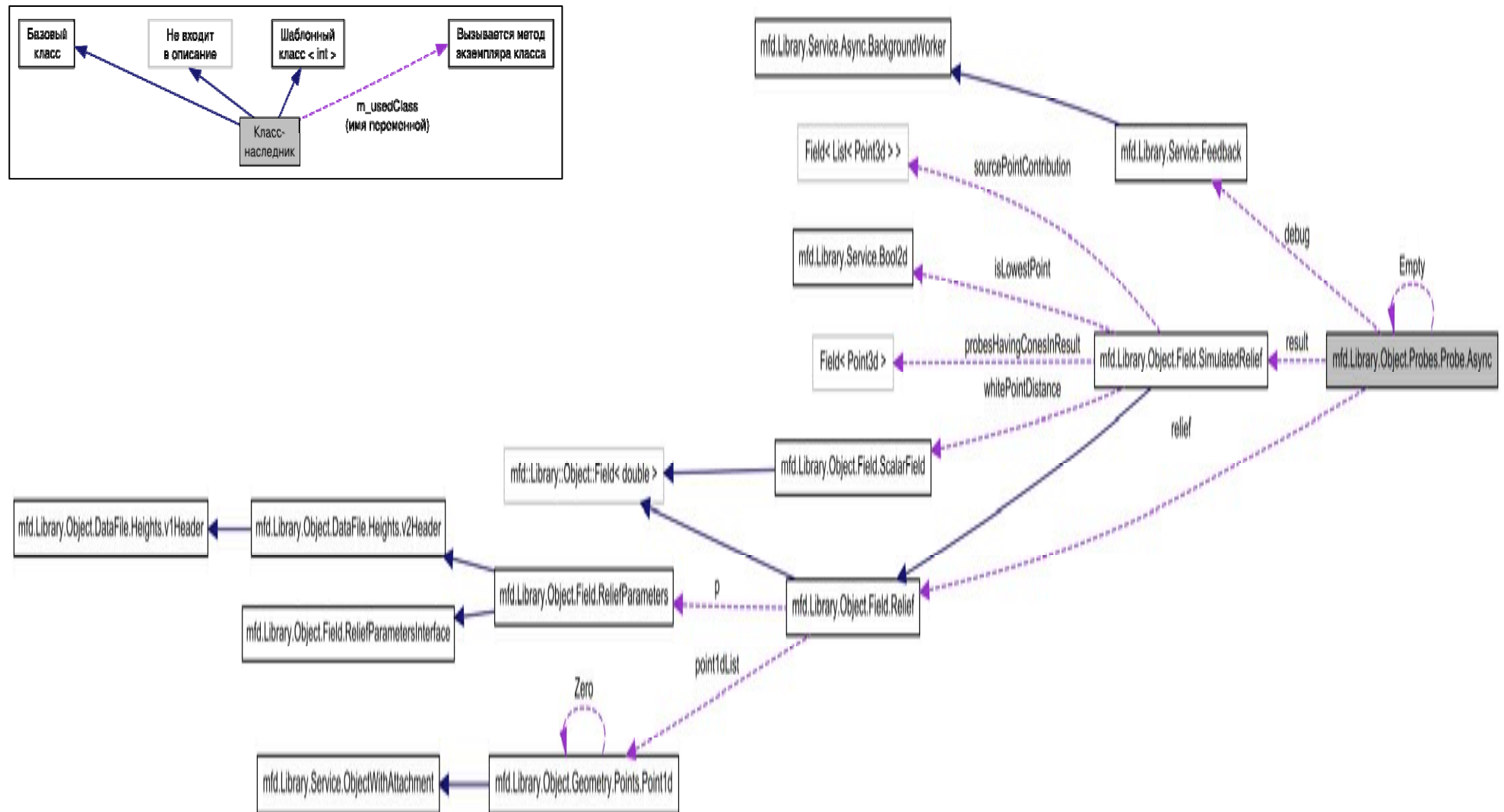


Рис. 4.4. Схема вызовов при организации многопоточности.

§6. Структура программных средств реализованного комплекса

Программные модули комплекса реализованы в виде байт-кода для виртуальной машины CLR (Common Language Runtime) и кода для API Win32. Средства управления работой программных модулей имеют консольный и оконный интерфейсы.

Комплекс спроектирован так, что допускается самостоятельное использование внутренней библиотеки `mfd.Library` отдельно от библиотек визуализации и пользовательского интерфейса. Указанная библиотека предназначена для загрузки данных, геометрической интерпретации измерений и моделирования рельефа поверхности. Кроме того, пользователю предоставлена возможность подключения своих библиотек для визуализации данных. По умолчанию программный комплекс использует собственную библиотеку `mfd.Gui.Windows`, которая опирается на `Windows.Forms` и `OpenGL`.

Для решения систем уравнений, программный комплекс использует оптимизированную версию библиотеки LAPACK в составе библиотеки классов `ILNumerics.Net` версии 1.4. Библиотека LAPACK содержит решатели основных задач линейной алгебры.

В библиотеке `mfd.Gui.Windows` для визуализации данных используется привязка стандарта `OpenGL 2.1` из кроссплатформенной библиотеки классов `Tao Framework` версии 2.1. Применяется связка органов управления `Tao Framework OpenGL` и `Windows.Forms`. Кроме того, в библиотеке `mfd.Library` реализован ограниченный вариант привязки `OpenGL` к `.Net Framework` (пространство имен `mfd.Library.Service.Graphics.OpenGL`).

В ряде случаев возникает необходимость использовать стороннюю реализацию `OpenGL`. Такие ситуации возникают при запуске программного комплекса в нестандартном окружении: использование вывода на дополнительный экран или работа в режиме клиента служб терминалов. Для этого наряду с реализацией, встроенной в операционную систему, допускается возможность использования альтернативных реализаций `OpenGL`. По умол-

чанию программный комплекс использует графическую библиотеку Mesa3D версии 7.6, которая программно реализует возможности, описываемые стандартом OpenGL 2.1.

Предусмотрена возможность локализации интерфейса на язык пользователя путем использования специально подготовленной библиотеки.

В программном комплексе предусмотрено сохранение журнала системных событий в файле процессе работы. В нем, кроме прочего, хранится информация об ошибках, возникающих при работе. Это позволяет упростить развертывание и поддержку программного комплекса.

Кроме того, хранится история изменений изображений. Пользователь имеет возможность просматривать список примененных преобразований (рис. 4.2, пп.8 и 9) и произвольно отменять произведенные преобразования, используя меню *Filtering* → *Undo*.

Основные операции в оконном интерфейсе осуществляются с помощью меню (см. рис. 4.2). Меню *File* отвечает за группу компонентов *Ввод файла* схемы на рис. 3.3. Пункт *File*→*Open* позволяет загрузить и преобразовать данные в виде изображения, либо в виде цифровой информации о сигнале. Меню *File*→*Save* позволяет сохранить данные в виде изображения, либо также в цифровом виде.

Меню *View* отвечает за способ отображения информации на экране. Подпункт *Display Mode* позволяет выбрать необходимый способ отображения. Пункт *View*→*Display Mode*→*Image*→*Switch to this view* позволяет посмотреть двумерное представление рельефа, закодированное с помощью выбранной палитры. Пункт *View*→*Display Mode*→*Profiles*→*Switch to this view* позволяет посмотреть профили рельефа поверхности. Это основной режим работы программного комплекса. Пункт *View*→*Display Mode*→*3D*→*Switch to this view* позволяет посмотреть трехмерное изображение рельефа поверхности. Подпункты меню *Show Grid* и *Solid Surface* переключают способ вывода трехмерной информации. Информация может выводиться либо в виде каркаса (тем самым позволяя наблюдать линии уровня), либо, соответственно, в

виде поверхности объекта, представляющего собой рельеф поверхности образца. Вывод трехмерного изображения поверхности был реализован с целью повышения удобства наблюдения нанообразований и структуры их расположения на ней.

Для обеспечения наглядности представления процесса подстройки туннельной оболочки, реализован алгоритм подсветки величины отклонения от соотношения (2.6). Для каждой точки исходного рельефа ищется расстояние от туннельной оболочки, построенной в данной точке, до моделированного рельефа поверхности. Цвет назначается в зависимости от полученного расстояния.

С помощью меню *Filtering*→*Noise* реализуется предварительная обработка сигнала. Пункт *Filtering*→*Noise*→*Average Out Brightness*→*Four Preceding Lines* осуществляет межстрочковую осуществить фильтрацию сигнала (см. гл. 1). Пункт *Filtering*→*Noise*→*Average Out Brightness*→*Standard* позволяет уменьшить уровень шумов.

С помощью меню *Filtering*→*Tunnel gap* реализована возможность задания величины туннельного зазора. Меню *Filtering*→*Restoration* отвечает за формирование модели рельефа по выбранной пользователем модели иглы. В программном комплексе проводится обработка рельефа поверхности с использованием 3D- и 2D- моделей игл. 2D-модель обладает тем преимуществом, что пользователю удобно анализировать точки контакта. Это связано с тем, что точка контакта в данном случае всегда расположена в плоскости выбранного профиля рельефа. Таким образом, по 2D-модели легко узнать, как работает 3D-модель.

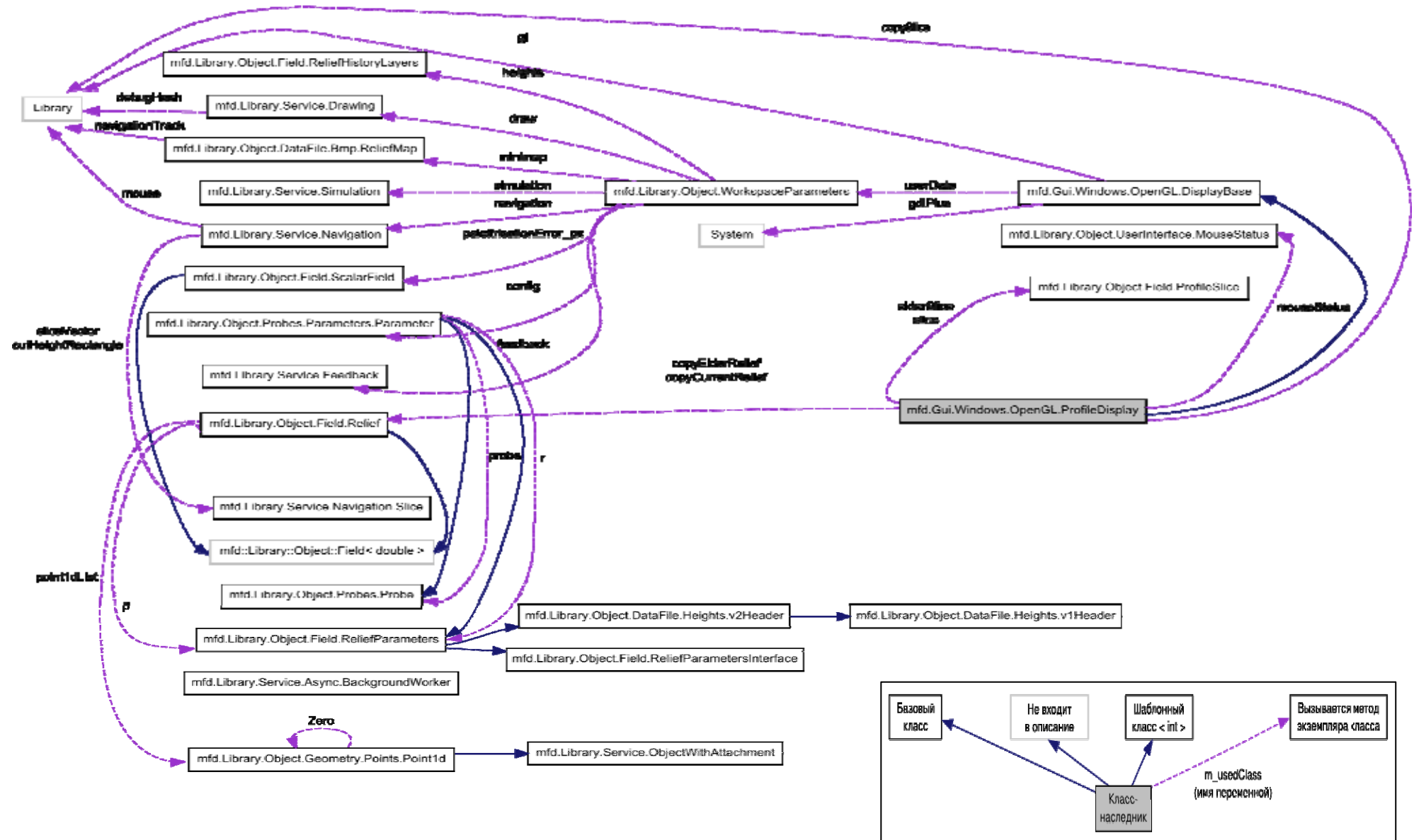


Рис. 4.5. Схема вызовов в режиме отображения профилей и принятые в схеме обозначения.

Меню *Filtering*→*Curr and Prev Relief Delta* запускает построение для каждого узла сетки значение разницы высоты рельефа поверхности до и после обработки, после чего работать с ним как с рельефом поверхности. Меню *Filtering*→*Relief slope* позволяет учесть наклон образца при сканировании. Подпункт *Suggest LSD Plane* запускает построение плоскости наклона, сумма квадратов отклонений от которой до каждой точки каркаса рельефа минимальна, и на наименее удалена по нормали. Подпункт *Suggest AvgAngle normal* позволяет вычислить диапазон векторов наклона к плоскости, который наиболее часто встречается. Из полученного диапазона выбирается вектор, соответствующий его середине. Подпункт *Draw Plane* строит требуемую плоскость по заданному пользователем каноническому уравнению плоскости. Меню *Filtering*→*Relief slope*→*Account Slope* реализует вычисления по устранению выбранного наклона путем проектирования каркаса рельефа на заданную плоскость наклона и интерполяции полученного каркаса рельефа.

На рис. 4.5 показаны классы, участвующие в графическом отображении рельефа поверхности на примере рисования профилей рельефа с помощью OpenGL. Класс `mfd.Gui.Windows.OpenGL.ProfileDisplay` содержит методы для инициализации, рисования, изменения размеров окна, отслеживания событий клавиатуры и мыши. Переменная `gl` его родительского класса содержит методы OpenGL, `gdiPlus` – методы GDI+, а `userData` – информацию о сеансе работы с программным комплексом.

§7. Методика применения программного комплекса

Сценарий обработки измерений задается с помощью иерархической системы меню, которая представлена в таблице 4.3. После открытия файла, в левой части окна программы будет выведен график первого профиля рельефа. При подведении курсора мыши к узлу профиля, показывается всплывающая подсказка с информацией о координатах указанного узла (показывается расстояние

по горизонтали от начала профиля до данной точки и высота точки в нанометрах — см. рис. 4.2).

В правой части окна выводится двумерное изображение рельефа поверхности. Оно раскрашено по высоте в соответствии с выбранной палитрой. Выбор палитры осуществляется через меню *View→Palette*.

Комплекс позволяет визуализировать профиль разреза рельефа по заданному пользователем направлению. Выбор линии осуществляется одним из трех способов:

1. Указанием в поле «Profile number» номера линии сканирования;
2. Выбором точки, через которую будет проходить разрез, на двумерном изображении рельефа поверхности.

Выбор направления разреза осуществляется с помощью контекстного меню. Оно открывается по щелчку правой кнопки мыши. С помощью пункта *Slice from here→Horizontal* можно указать на рисование профиля вдоль линии сканирования, на которой находился курсор мыши в момент вызова контекстного меню. Пункт *Slice from here→Vertical* позволяет выбрать указание рисовать профиль поперек линии сканирования.

3. Путем указания концов линии разреза рельефа поверхности.

При нахождении мыши в пределах двумерного изображения рельефа и нажатии ее левой кнопки, появляется отрезок линии среза. При отпускании левой кнопки мыши, строится профиль рельефа.

Визуализация профиля разреза рельефа по заданному направлению позволяет исследовать форму нанообразований на рельефе поверхности. Рис. 4.2 иллюстрирует построение горизонтальной линии разреза. Линия разреза показана на двумерном изображении рельефа.

Для замены модуля, описывающего модель острия зонда (см. гл. 3), должен быть сформирован новый файл, представляющий собой библиотеку классов с именем `CustomProbeParameters.dll`. Этот файл должен содержать класс `CustomProbeParameters`, являющийся наследником класса `Parameter`, описанного ниже.

```

1 namespace mfd.Library.Object.Probes.Parameters {
2 public abstract partial class Parameter {

3     protected double curvatureRadius_px;
4     protected ReliefParameters r;

5     protected abstract ProbeSimulation CreateProbeSimulation();
6     public abstract void CopyFrom(Parameter source);
7     public abstract double z(double probePositionX, double
        probePositionY, double probePositionZ, double x, double
        y);

8     public virtual int MaximalRadius_px; get;
9 } }

```

В конструкторе класса пользователь должен присвоить переменной `curvatureRadius_px` (строка 3) значение радиуса закругления модели острия в единицах измерения, равных расстоянию между строками сканирования (см. описание формата XML в гл. 3). Это значение используется программным комплексом в дальнейших вычислениях. Конструктор класса должен принимать всего 2 параметра – переменную типа `double`, в которую должна передаваться величина радиуса закругления, и `ReliefParameters`, содержащую метрические характеристики рельефа поверхности, обрабатываемого в момент создания экземпляра класса. В процессе работы конструктора переменная `r` (строка 4) должна получить переданное в него значение.

От пользователя требуется предоставить реализацию метода `CreateProbeSimulation` (строка 5) должен возвращать экземпляр класса `mfd.Library.Target.Signal.Filtering.Simulation.Probes.ProbeSimulation`, в котором описывается способ проведения физических расчетов. Обычно достаточно возвращать `new ProbeSimulation(this)`. При необходимости изменения физической модели, пользователь должен возвращать экземпляр

своего класса-наследника `ProbeSimulation`, реализующего вычисления в соответствии с новой моделью.

За задание геометрии острия зонда отвечает метод `z` (строка 7). В качестве параметров ему передаются три координаты x_0, y_0, z_0 точки, в которой располагается кончик зонда, и абсолютные координаты по осям $0x$ и $0y$ точки, для которой требуется вернуть величину координаты по оси $0z$ (в соответствии с тем, какой функцией описывается поверхность модели острия зонда). Такое задание координат фактически означает, что модель острия задается в виде функции от двух аргументов x и y : $z = f(x_0, y_0, z_0; x, y)$.

При отличии максимальной ширины модели зонда от удвоенного значения `curvatureRadius_px`, пользователь должен переопределить свойство `MaximalRadius_px`, возвращая максимальное расстояние от проекции на плоскость $0xy$ вершины модели зонда до границы ее проекции на эту же плоскость.

Метод `CopyFrom` (строка 6) должен заполнять содержимое экземпляра пользовательского класса в соответствии со значениями полей объекта, передаваемого в качестве параметра.

Таблица 4.3. Основные пункты меню программного комплекса.

File		
	Open...	
	Save As	
	Exit	
View		
	Display Mode	
	Image	Switch to this View
	Profiles	Switch to this View
	3D	Switch to this View
		Draw Grid
		Solid Surface

Таблица 4.3. Основные пункты меню программного комплекса.

Filtering	Palette	PaletteGray	
		PaletteRanbow	
		PaletteNanotechnology1997	
		...	
	Scan Information...		
	Undo	...	
	Noise		
		Average Out Brightness	
			Standard
			Four Preceding Lines
Service	Restoration	Using Probe Simulation	
			Spherical...
			Cylinder-cone-sphere...
			Cone-sphere...
			Cylinder-circle (2D)...
	Curr and Prev Relief Delta		
	Relief Slope		
		Account Slope...	
		Draw Plane...	
		Suggest LSD Plane...	
Service	Tunnel gap...	Suggest AvgAngle normal...	
	Save delta values log...		
	Edit configuration file		
	View probe simulation parameters		



Рис. 4.6. Окно открытия файла.

Parameters Input Window

FieldName	FieldValue
width_nm	20.06
length_nm	0
height_nm	20
scanLineCount	256
width_px	256

Cancel OK

Рис. 4.7. Окно ввода метрических характеристик рельефа поверхности.

После открытия файла и просмотра первоначального изображения рельефа поверхности, целесообразно произвести начальную обработку данных встроенными средствами: удаление шумов и устранение флуктуаций яркости по строкам. Для удаления шумов служит меню *Filtering→Noise→Average Out Brightness→Standard*. Для межстрочного осреднения необходимо воспользоваться пунктом меню *Filtering→Noise→Average Out Brightness→Four Preceding Lines* (см. описание рис. 4.2).

После задания туннельного зазора указывается модель туннельной оболочки и ее параметры. Для этой цели используется пункт меню *Filtering→Restoration→Using probe simulation*. В открывшемся подменю следует выбрать нужный тип туннельной оболочки. Окно параметров модели показано на рис. 4.9. Описания моделей игл и соответствующих им пунктов меню приведены в §1.

В начале работы с новым рельефом, полезно сначала воспользоваться двумерной моделью туннельной оболочки. По рельефу, полученному с помощью двумерной туннельной оболочки, легче отслеживать происходящие изменения. Данный факт вызван тем, что взаимодействие иглы и поверхности в соответствии с двумерной моделью имеет место только в пределах одной линии сканирования. За двумерные модели отвечают подпункты меню *Parabolic (2D)* и *Cylinder-circle (2D)*.

В случае трехмерной модели оболочки, игла с острием, находящимся в плоскости данного профиля рельефа, может взаимодействовать с рельефом не в центральном сечении. Поэтому задача оценки взаимодействия представляет очевидную сложность. Выбор трехмерной модели туннельной оболочки осуществляется с помощью подпунктов меню *Spherical*, *Cone-sphere* или *Cylinder-cone-sphere*.

После подтверждения размеров туннельной оболочки в окне выбора параметров модели и проверки соответствия условиям сканирования, начинается моделирование рельефа поверхности для выбранной модели туннельной оболочки. Процесс моделирования выполняется многопоточно в фоновом режиме.

Ход процесса отображается на индикаторе прогресса (рис. 4.10). При необходимости прервать процесс моделирования, следует нажать кнопку остановки



, расположенную справа от индикатора.

Для трехмерной модели иглы осуществляется расчет параметров, необходимых для оценки качества. В каждой точке исходного рельефа ищется расстояние от туннельной оболочки до смоделированного рельефа поверхности. Цвет точки автоматически назначается в соответствии найденной величиной: бирюзовым цветом помечаются точки, в которых расстояние превышает 1,5 Å; зеленым цветом — 3 Å; желтым цветом — 4,5 Å; малиновым цветом помечаются точки с нулевым расстоянием равно нулю. Таким образом, малиновым цветом помечаются основания туннельных оболочек, которые хотя бы одной своей точкой внесли вклад в результат. Соответствие раскраски расстояниям может быть изменено пользователем с помощью файла конфигурации (см. §8).

При выборе двумерной модели, точки полного соответствия соотношению также помечаются малиновым цветом. Точки, в которых оно не выполнено, помечаются желтым цветом. Пользователь может включить рисование туннельной оболочки (пункт Highlight probe в правой части вкладки Display) и просмотреть, каким образом игла взаимодействовала с рельефом. Данная возможность доступна и для трехмерных моделей, но поиск сечения, в котором вместе находятся и острие иглы, и точка контакта, требует определенного навыка.

Если на изображении профилей находится много точек, соответствующих нулевому расстоянию, то нужно отменить текущее моделирование (меню *Filtering*→*Undo*) и повторить его с увеличенным радиусом туннельной оболочки. Если же уровень соответствия указанному выше соотношению низок, то нужно аналогично повторить моделирование с уменьшенным радиусом.

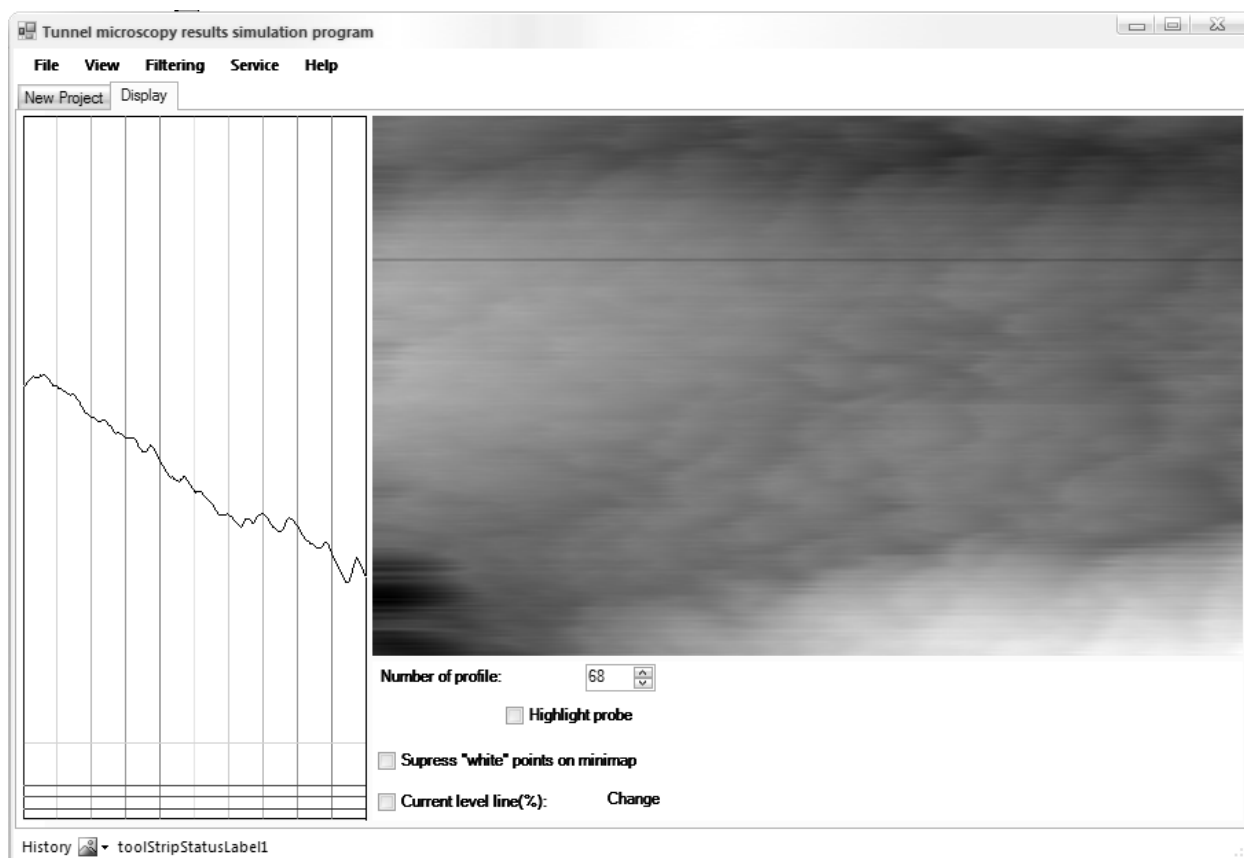


Рис. 4.10. Окно программного комплекса в процессе работы с профилями рельефа поверхности меди.

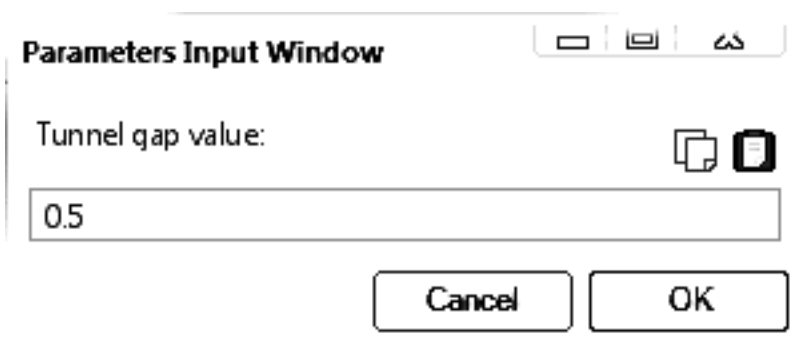


Рис. 4.8. Окно задания величины туннельного зазора.

Parameters Input Window

Enter the angular opening of cone, °:

Enter the maximal width of the probe, nm:

Enter the curvature radius, nm:

Buttons: Cancel, OK

Рис. 4.9. Окно ввода параметров туннельной оболочки.



Рис. 4.10. Индикатор прогресса и кнопка остановки.

Parameters Input Window

Plane equation:

Buttons: Cancel, OK

Рис. 4.11. Окно ввода уравнения плоскости.

С целью улучшения детализации изображения, пользователь имеет возможность устранить наклон поверхности. Для этого сначала автоматически или вручную подбирается плоскость, обеспечивающая наилучший результат. В программном комплексе реализованы несколько алгоритмов определения наклона: вычисление плоскости, имеющей минимальную сумму квадратов расстояний от точек до плоскости; определение вектора наклона, который чаще всего встречается. Эти алгоритмы доступны по пунктам меню *Filtering*→*Relief slope*→*Suggest LSD Plane* и *Filtering*→*Relief slope*→*Suggest AvgAngle normal*. После получения компонентов вектора нормали к плоскости наклона, пользователь осуществляет устранение наклона выбором пункта меню *Filtering*→*Relief slope*→*Account Slope* (рис. 4.11). В окно требуется ввести компоненты вектора $\{A, B, C\}$ нормали к плоскости, на которую будет осуществляться проектирование. Нормирование вектора осуществляется программой автоматически.

Процесс поиска плоскости наклона и устранения наклона отображается с помощью индикатора прогресса. Отмена вычислений производится таким же способом.

Результатом обработки данных сканирования является восстановленный рельеф поверхности и наиболее вероятные параметры иглы.

На рис. 4.12 показаны результаты моделирования рельефа поверхности (пунктирная линия со штриховкой) для цилиндрической иглы с разными радиусами закругления острия. Измеренная линия рельефа (траектории движения иглы при сканировании) показана сплошной линией.

Непосредственно из рисунка видно, что если радиус закругления выбран слишком большим (a), то нарушается условие непрерывности контакта туннельной оболочки и модели рельефа (2.6) в большинстве точек измерения. Такие точки помечены желтым цветом. В случае меньшего радиуса закругления (b), неравенство выполняется почти во всех узлах сетки измерений.

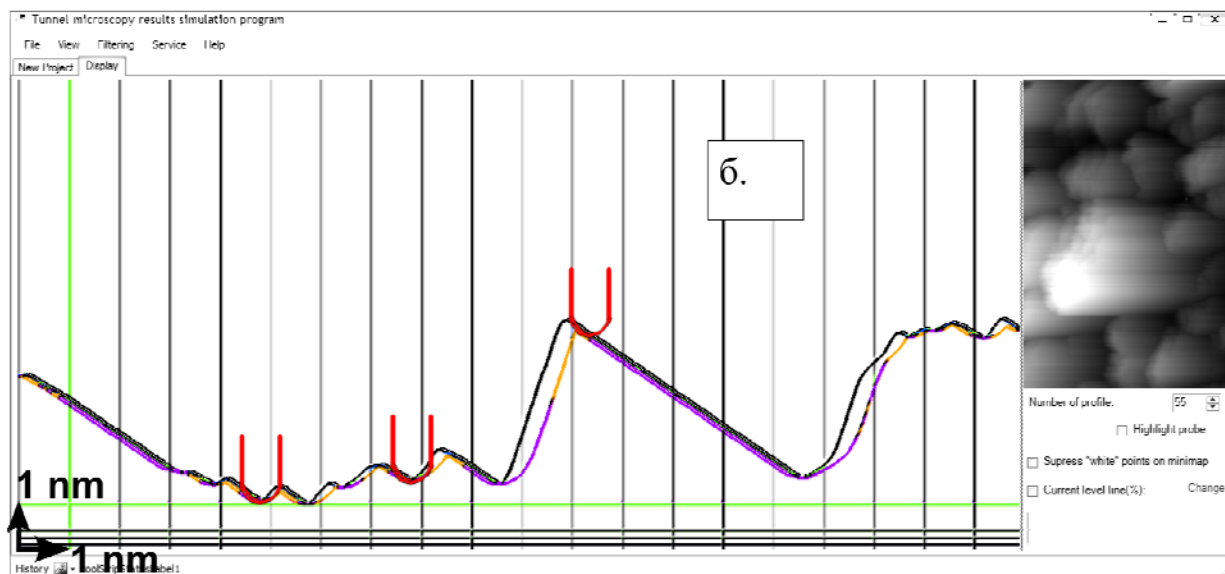
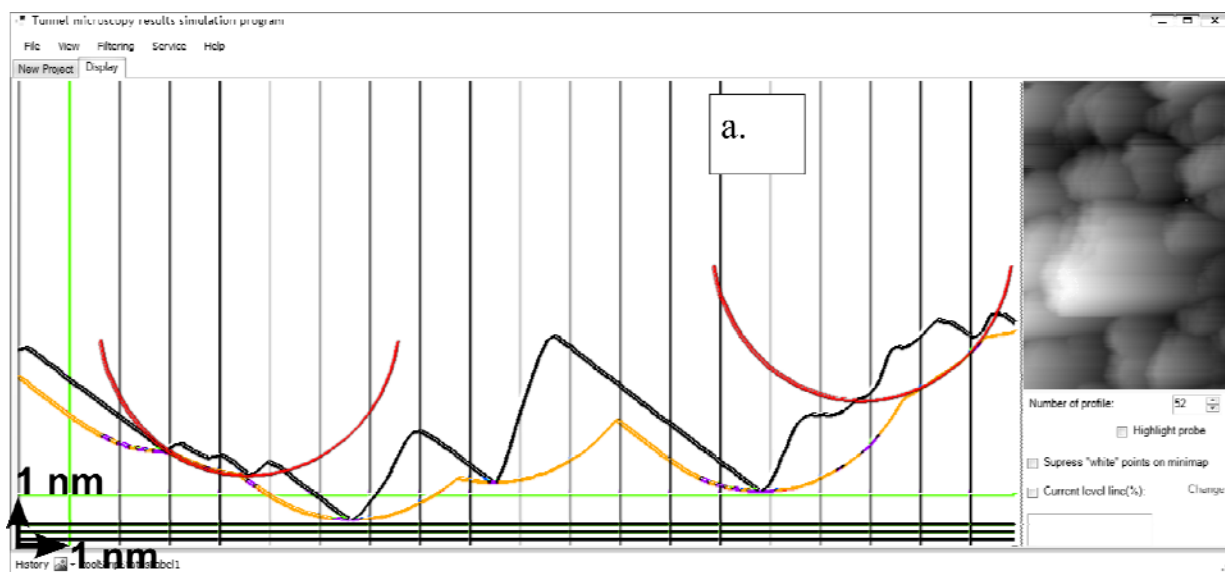


Рис. 4.12. Моделирование рельефа поверхности
для разных радиусов закругления острия.
а. радиус закругления 3 нм, б. 0,5 нм.

§8. Средства настройки программного комплекса

В программном комплексе реализована возможность изменения системных настроек, используемых по умолчанию. На рис. 4.13 показано окно настройки. Данное окно вызывается командой Service→Edit configuration file. Окно открывается в виде отдельной вкладки. В левой части окна записано содержимое конфигурационного файла в формате XML. Настройкам соответствует адрес XPath

/configuration/appSettings/add,

причем в элементе *add* атрибут *key* означает название настройки, а *value* – ее значение. В таблице 4.4 указаны различные настройки и возможные значения.

При нажатии на командную кнопку «Apply changes», происходит сохранение файла конфигурации, закрытие вкладки, применение изменений. При нажатии на командную кнопку «Cancel and return», происходит отмена изменений файла конфигурации и закрытие вкладки. Независимо от нажатой кнопки, осуществляется переключение в ту вкладку, в которой пользователь до этого работал.

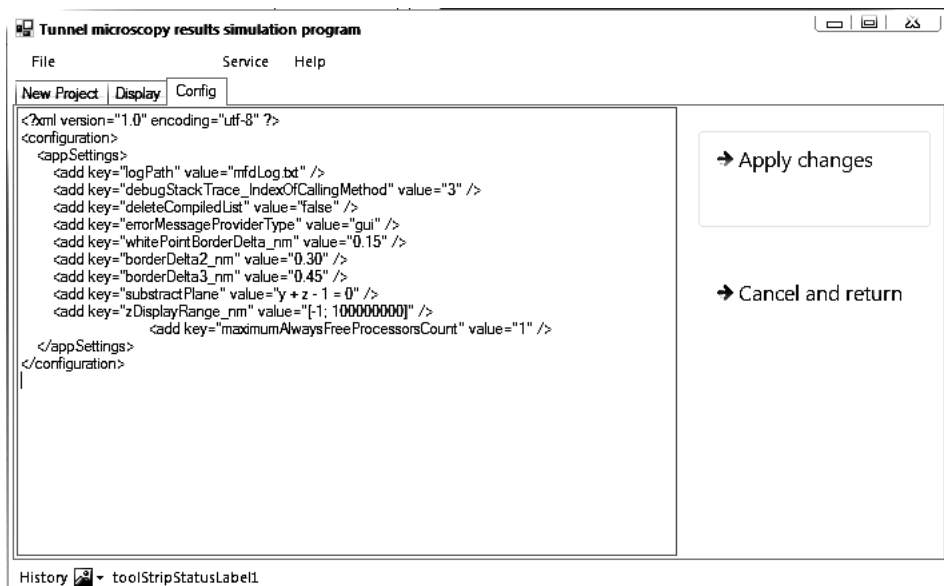


Рис. 4.13. Окно настройки программного комплекса.

Таблица 4.4. Параметры конфигурации.

Название	Описание	Значения
logPath	Путь к файлу, в котором ведется журнал ошибок	mfdLog.txt
debugStackTrace_IndexOfCallingMethod	Индекс вызывающего метода в трассировке стека при печати сообщения об ошибке	3 для версии Microsoft .NET Framework 2.0.50727
errorMessageProviderType	Способ вывода сообщений об ошибках	gui (окно с сообщением), console (вывод в stdout), none (отключить вывод сообщений об ошибках)
deleteCompiledList	Управление дисплейными списками OpenGL. Указывает, удалять ли дисплейный список после использования	false (не удалять) true (удалять)
maximumAlwaysFreeProcessorsCount	Максимальное количество процессоров, которые необходимо держать свободными от нагрузки	Обычно 1 (по причине необходимости резервирования процессора на нужды пользовательского интерфейса и сборщика мусора)
whitePointBorderDelta_nm	Граница №1 расстояния от рельефа до туннельной оболочки (см. выше), нм	0.15
borderDelta2_nm	Граница №2 расстояния, нм	0.30
borderDelta3_nm	Граница №3 расстояния, нм. По умолчанию, после нее точки помечаются желтым цветом	0.45
subtractPlane	Уравнение плоскости, которое по умолчанию выводится при вызове функции учета плоскости наклона	$y + z - 1 = 0$
zDisplayRange_nm	Область отображения значений по высоте. Если высота не входит в интервал, значение не показывается на графиках	[-1; 10000]

§9. Использование комплекса в исследованиях рельефа поверхности

Разработанный программный комплекс использовался для интерпретации результатов измерений поверхностей различных металлов в исследованиях, выполненных сотрудниками РХТУ им. Д.И. Менделеева в 2005 – 2009 годах. Измерения проводились в воздушной среде с помощью туннельного микроскопа НПО «Алмаз». Исследовались поверхности платины, вольфрама, молибдена и меди, нанесенные на стеклянную подложку путем вакуумного напыления. В общей сложности было получено около 200 изображений с пространственным разрешением от 0,2 нм до 1 нм и размером изображения от 128x128 точек до 512x512 точек.

Наиболее часто использовался режим сканирования с напряжением в туннельном зазоре 1 В и токе около 1 нА. Описанные во второй главе расчеты показывают, что туннельный зазор составляет примерно 0,5 нм.

Рассмотрим рисунки с примерами изображений рельефа различных поверхностей до и после учета геометрии иглы (рис. 4.14, 4.16).

Непосредственно из рисунков следует, что учет формы и размеров острия иглы при интерпретации измерений приводит к существенным изменениям изображения поверхности. Рассмотрим наиболее заметные из них.

Впадины становятся шире, выступы уже. Наиболее заметные изменения претерпевают изображения мелких образований. Этот вывод находится в хорошем соответствии с результатами исследования, выполненными в главе 2.

Преобразованное изображение очищается от различных шумов. Шумы на исходном изображении проявляются в виде мелкой ряби, которая заметна практически на всех образованиях на поверхности.

Исчезают также некоторые артефакты: кажущиеся нанообразования, «гребёнка», похожая на интерлейсинг (она хорошо заметна внизу рис. 4.14 в). Некоторые из них достаточно сложно удалить с изображения традиционными методами, хотя визуально они действительно воспринимаются скорее как несуществующие объекты. Отсутствие их при учете геометрии острия иглы связано с тем, что их существование противоречит физическим принципам скани-

рования. Действительно, значительная часть артефактов исчезает даже при малой величине радиуса закругления острия иглы. На рис. 4.14 д показано изображение, демонстрирующее степень изменения рельефа поверхности в процессе преобразования.

Существенно изменяется форма образований нанометровых размеров. На исходных изображениях они имеют сферическую форму (рис. 4.14 а). При интерпретации измерений с учетом геометрии иглы, их форма становится похожей на пирамиду (рис. 4.14 б). Изменения нетрудно заметить, наблюдая профили на вертикальных срезах.

На изображении можно наблюдать сглаженные ребра поверхности образований (рис 4.14, 4.15). На исходном изображении изгибы не видны, в силу того, что при огибании иглой углов радиус кривизны траектории движения иглы равен радиусу закругления острия.

На рис. 4.14 з показано изображение платины (рис. 4.14 в) после обработки. Использовалась модель иглы радиусом 1,5 нм. На рис. 4.14 з приведено изображение разности исходного и обработанного рельефов поверхности. Желтые и красные оттенки соответствуют областям наибольших изменений.

Учет формы острия иглы приводит к тому, что впадины малого размера приобретают округлые края (рис. 4.16). Причина таких изменений состоит в том, что игла не в состоянии проникнуть в вершину угла, как это показано на рис. 4.17.

После учета геометрии острия иглы становятся легко различима структура расположения малых образований (рис. 4.16). При внимательном рассмотрении можно заметить, что она прослеживается и на исходном изображении. Лучшая видимость ребер связана с уменьшением их размеров.

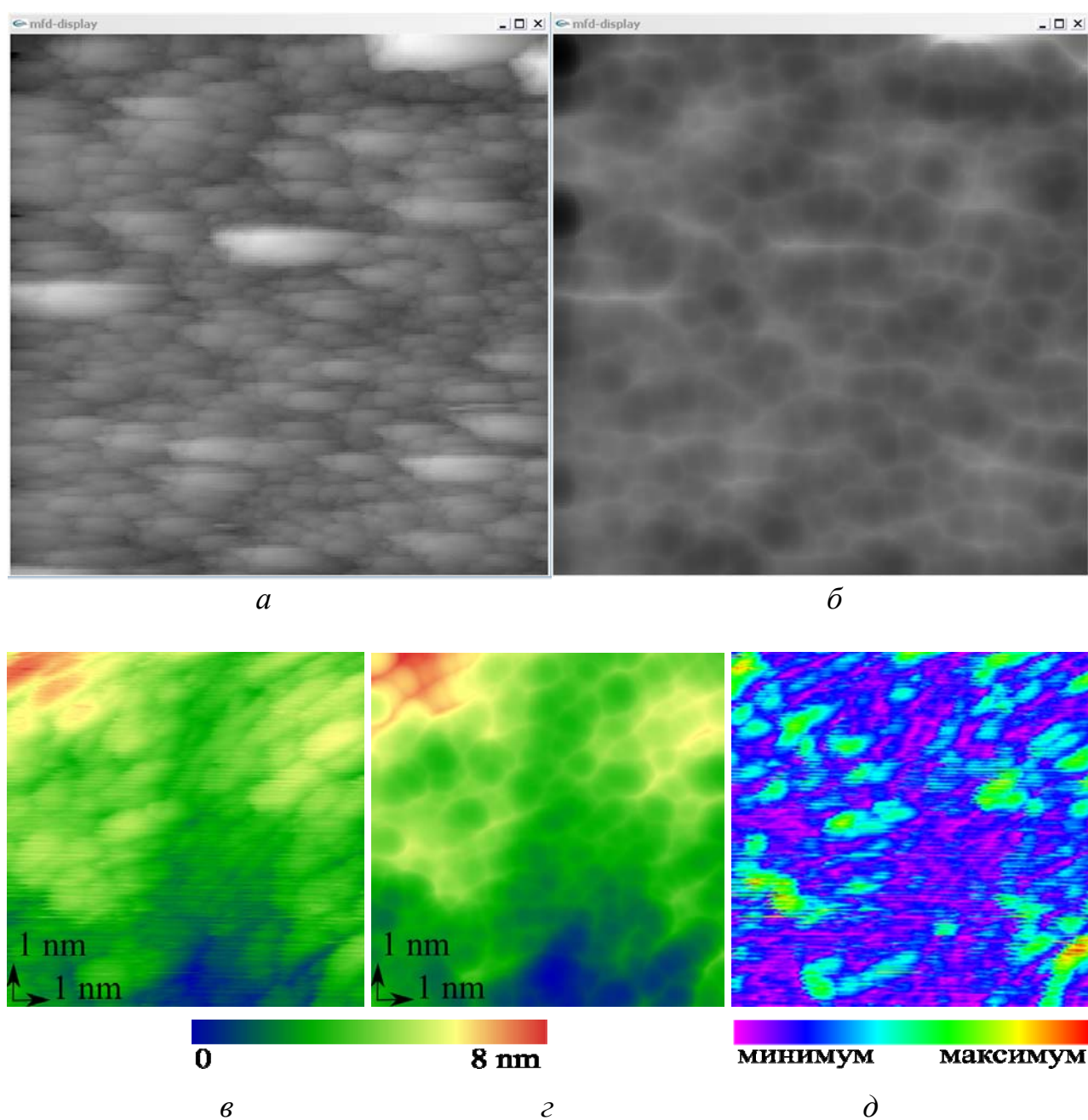


Рис. 4.14. *a* – изображение поверхности меди (размер 256x256 нм);
б – изображение, полученное при учете геометрии острия иглы;
в – поверхность платины на подложке размером 30x30 нм;
г – изображение после обработки,
д – разница между исходным и обработанным изображениями.

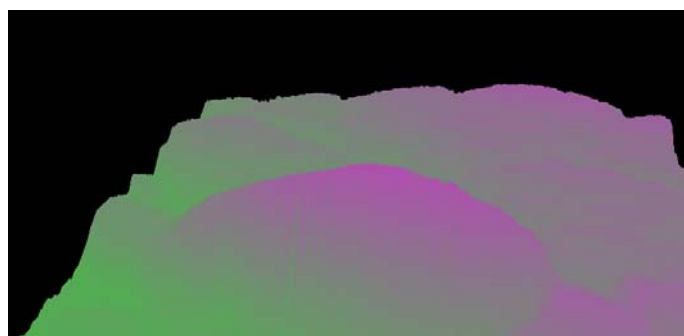
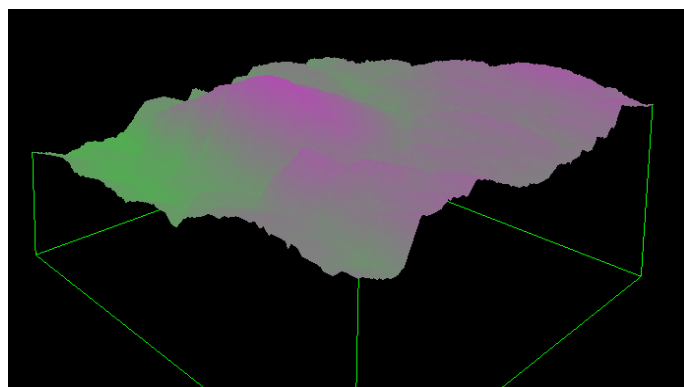
*a**б*

Рис. 4.15. *a* - нанообразование в виде трехгранного кристалла (видна вершина и два ребра, размер основания равен 20 нм); *б* – тот же объект под другим углом зрения на фоне рельефа поверхности.

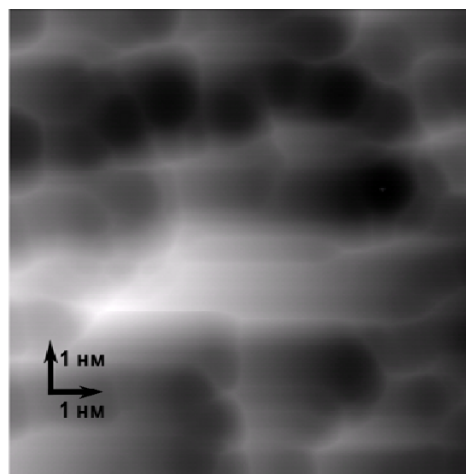
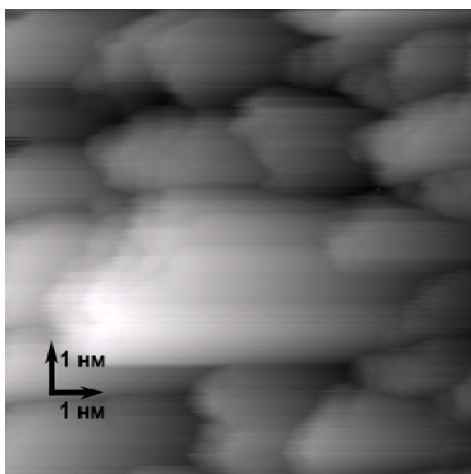


Рис. 4.16. Участок поверхности платины размером 10x10 нм, полученный с помощью иглы с диаметром острия 1,5 нм.
a – при учете радиуса закругления острия, *б* – без учета.
 На левом изображении хорошо видны овальные впадины и форма поверхности нанобразований.

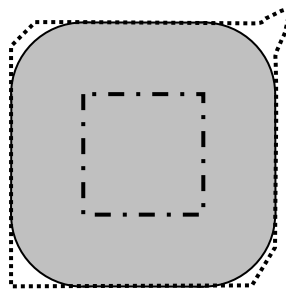


Рис. 4.17. Пунктирная линия - контур реального рельефа;
 штрих-пунктирная линия - измеренный контур, который виден на
 изображении, построенном по измерениям;
 сплошная линия – восстановленный контур, учитывающий размеры
 острия.
 Серым цветом окрашено видимое изображение рельефа,
 белым - невидимые из-за размеров острия участки рельефа.

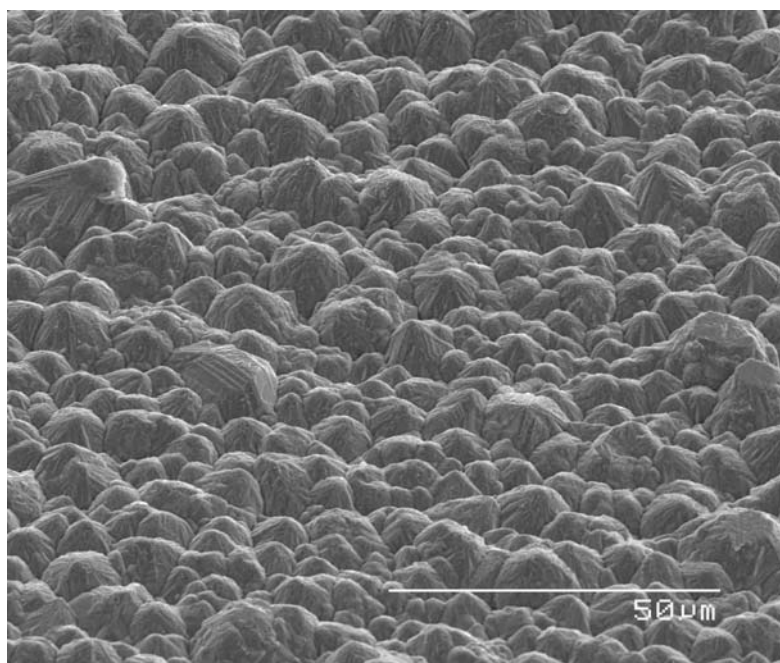


Рис. 4.18. Изображение поверхности меди, нанесенной вакуумным
 напылением на стеклянную подложку.

При учете радиуса закругления острия иглы заостряются вершины малых образований и, вследствие этого уменьшается отношение ширины ребра к размеру малого образования. Это увеличивает локальную контрастность изображения и приводит к улучшению различимости изгибов поверхности. Рассматриваемое изображение интересно сравнить с изображением микрокристаллов меди на стеклянной подложке (рис. 4.18), сфотографированного с помощью электронного микроскопа.

Видимое на фотографии разрешение в 1000 раз хуже, чем при использовании туннельного микроскопа. В некотором смысле можно сказать, что туннельный микроскоп позволяет увидеть рельеф поверхности на гранях кристаллов, которые изображены на снимке.

Следует заметить, что отмеченные изменения в изображениях, которые возникают вследствие учета геометрии острия иглы, сохраняются при варьировании размера острия иглы в некотором интервале. Это дает основание считать, что наблюдаемые особенности рельефа поверхности соответствуют реальности.

Выводы четвёртой главы

1. Проведенные исследования показали, что программный комплекс позволяет достаточно быстро получить надежные оценки формы и размеров острия иглы.

2. Учет геометрии при интерпретации измерений приводит к существенным изменениям изображений поверхности. В результате оказывается возможным наблюдать важные детали рельефа исследуемой поверхности, некоторые из которых на исходном изображении различимы с трудом.

3. Учет геометрии иглы позволяет уменьшить количество шумов и артефактов.

Заключение

1. Создан алгоритм учета формы и размеров иглы туннельного микроскопа путем интерактивной обработки информации о процессе сканирования нанорельефа исследуемого образца. Показано, что программная модель для реализации алгоритма не может быть сведена к модели графических фильтров, обеспечивающих обработку информации путем применения функциональных преобразований.

3. Разработаны способы обеспечения многопоточности обработки информации и модификации программного кода модулей для программного комплекса зондовой микроскопии при ограничениях на сложность и трудоемкость реализации.

4. Реализован программный комплекс и продемонстрирована его эффективность в исследованиях нанорельефа поверхности туннельным микроскопом. На примерах обработки реальных измерений рельефа поверхностей различных веществ продемонстрирована возможность использования программного комплекса для уменьшения количества артефактов, связанных с шумами, погрешностями измерений и особенностями процесса сканирования. Эффективность достигаемых при этом результатов основывается на учете физической модели взаимодействия иглы и поверхности.

Литература

1. В.Л. Миронов. Основы сканирующей зондовой микроскопии. М.: Техносфера, 2004.
2. А.С. Филонов, И.В. Яминский. Руководство пользователя пакета программного обеспечения для управления сканирующим зондовым микроскопом и обработки изображений «ФемтоСкан Онлайн». Версия 2.2.143. М: Центр перспективных технологий, 2008.
3. NanoEducator. Модель СЗМУ-Л15. Базовый прибор для научнообразовательного процесса в области нанотехнологии // Рук-во польз. М.: НИИФП, ЗАО «НТ-МДТ», 2005.
4. Лабораторный нанотехнологический комплекс «УМКА» / Концерн «Наноиндустрия». Дата обновления: 27.05.2010. URL: <http://www.nanotech.ru/pages/about/umka.htm> (дата обращения: 30.09.2010).
5. Н. Г. Циркунова, В. Е. Борисенко. Геометрия зондовой микроскопии: влияние радиуса закругления зонда на точность формируемого изображения // VII межд. сем. "Методологические аспекты сканирующей зондовой микроскопии" (БелСЗМ). Минск, 2006.
6. Akkana Peck. Beginning GIMP: from novice to professional. USA, NY: Springer-Verlag New York Inc., 2006.
7. R.V. Gainutdinov, P.A. Arutyunov. Artifacts in Atomic Force Microscopy // Russian Microelectronics. 2001. Vol. 30, №4.
8. М.О. Галлямов, И.В. Яминский. Сканирующая зондовая микроскопия: основные принципы, анализ искажающих эффектов. Дата обновления: 08.09.2006. URL: http://spm.genebee.msu.ru/members/gallyamov/gal_yam/gal_yam1.html (дата обращения: 30.09.2010).
9. М.М. Горбунов-Посадов. Расширяемые программы. М.: Полиптих, 1999.
10. Технология открытых систем. [под ред. А.Я. Олейникова]. М.: Янус-К, 2004.
11. А.К. Платонов, В.В. Карташев. Методы обработки изображения рельефа поверхности в зондовой микроскопии: обзор программного обеспечения сканирующих зондовых микроскопов // Отчет ИПМ им. М.В. Келдыша РАН. 2009.

№5-028-09.

12. И.А. Дорофеев, В.Я. Косыев, А.А. Петрухин. Динамика туннельного контакта при импульсном повышении напряжения между иглой и образцом в туннельном микроскопе // Журнал технической физики. 1998. Т. 68, № 2.
13. J. Tukey. Exploratory Data Analysis. USA, California, Menlo Park: Addison-Wesley, 1977.
14. А.С. Филонов, И.В. Яминский. Обработка и анализ данных в сканирующей зондовой микроскопии: алгоритмы и методы // Наноиндустрия. 2007. №2 – Наноматериалы.
15. А.В. Давыдов. Цифровая обработка сигналов: тематические лекции. Екатеринбург: УГГУ, ИГиГ, ГИН, Фонд электронных документов, 2005.
16. Е.И. Шишкин, Е.В. Николаева. Исследование наноматериалов методами сканирующей зондовой микроскопии: руководство к лабораторным и практическим занятиям. Екатеринбург: 2008. (Уральский государственный университет им. А.М. Горького)
17. Б.А. Грибков. Сканирующая зондовая микроскопия поверхностной шероховатости и магнитных наноструктур: Автореф. дис. канд. физ.-мат. наук. – Н.Н., 2006.
18. Биомедицинские приложения сканирующей зондовой микроскопии: наноскопия, нанодиагностика и нанотехнология // Российские нанотехнологии. М.: Парк-медиа, 2008. Т. 3, № 3-4.
19. В.И. Балыкин, П.Н. Мелентьев. Нанолитография методами атомной оптики // Российские нанотехнологии. М.: Парк-медиа, 2009. № 7-8.
20. В.Ю. Юров, А.Н. Климов. Восстановление истинного СТМ-изображения поверхности с учетом дрейфа, наклона образца и калибровки керамики СТМ. // Труды ИОФАН. Проблемы нелинейных измерений микрообъектов в нанометровом и субмикронном диапазонах. М.: Наука, 1995.
21. Т. А. Поплевкин, А. А. Ерофеев. Математическая модель пьезоэлектрического двигателя с учетом гистерезиса // Применение туннельной микроскопии: межвуз. сб. науч. тр. М: МИРЭА, 1989.

22. Д. Калинкина, Д. Ватолин. Проблема подавления шума на изображениях и видео и различные подходы к ее решению // Компьютерная графика и мультимедиа. 2005. Т. 9, №2. Дата обновления: 23.05.2005. URL: <http://cgm.computergraphics.ru/content/view/74> (дата обращения: 30.09.2010).
23. В.А. Карташев, В.В. Карташев. Исследование движения иглы туннельного микроскопа относительно поверхности // Нано и микросистемная техника. М.: Новые технологии, 2008. №6.
24. Three-dimensional probe and surface reconstruction for AFM using a deconvolution algorithm / Anastas Bukharaev, Nikolai Berdunov, Denis Ovchinnikov, Kev Salikhov // Scanning Microscopy. 1998. Vol. 12, № 1.
25. Blind reconstruction of scanning probe image data / P. M. Williams, K. M. Shakesheff, M. C. Davies, D. E. Jackson, C. J. Roberts, S. J. B. Tendler // Journal of Vacuum Science and Technology. 1992. B 14 (2).
26. J. S. Villarrubia. Algorithms for scanned probe microscope image simulation, surface reconstruction, and tip estimation // Journal of Research of the National Institute of Standards and Technology. 1997. Vol. 102, № 4.
27. R. Chicon, M. Ortuno, J. Abellan. An algorithm for surface reconstruction in Scanning Tunneling Microscopy // Surface Science. 1987. Vol. 181.
28. Tip Characterization and Surface Reconstruction of Complex Structures with Critical Dimension AFM / G. Dahlen, M. Osborn, N. Okulan, W. Foreman, A. Chand, J. Foucher. Veeco Instruments Inc, 2004.
29. D. Tranchida, S. Piccarolo, R. A. C. Deblieck. Some experimental issues of AFM tip blind estimation: the effect of noise and resolution measurement // Institute of Physics Electronic Journal. 2006. Issue 10.
30. D. Keller. Reconstruction of STM and AFM images distorted by finite-size tips // Surface Science. 1991. Vol. 253, Issues 1-3.
31. E. Strassburg, A. Boag, Y. Rosenwaks. Reconstruction of Electrostatic Force Microscopy Images // Review of Scientific Instruments. 2005, № 76.
32. Identification of Size Differences of Gold Nanoparticles on Cell Surface by Curvature Reconstruction Method Using Atomic Force Microscopy / Hyonchol Kim,

- Koudai Oikawa, Naoya Watanabe, Masatsugu Shigeno, Yoshiharu Shirakawabe, Kenji Yasuda // *Japanese Journal of Applied Physics*. 2007. №46.
33. С.Ю. Васильев, А.В. Денисов. Особенности туннельно-спектроскопических измерений в конфигурации воздушного сканирующего туннельного микроскопа // *Журнал технической физики*. 2000. Т. 70, №. 1.
34. В. Г. Лифшиц. Современные приложения сканирующей туннельной микроскопии для анализа и модификации поверхности // *Соросовский образовательный журнал*. 2001. Т. 7, №5.
35. Д.А. Усанов, Р.К. Яфаров. Исследование поверхности материалов методом сканирующей туннельной микроскопии: уч. пос. для студ. факультета нано- и биомедицинских технологий. Саратов: Изд-во Саратов. ун-та., 2007.
36. G. L. Timp. *Nanotechnology*. USA, NY: Springer-Verlag New York Inc., 1999.
37. The electronic properties of grapheme / A.H. Castro Neto, F. Guinea, N.M.R. Peres, K.S. Novoselov, A.K. Geim // *Reviews of Modern Physics*. 2009. Vol. 81.
38. Т. Эрдеи-Груз. Основы строения материи. М.: Мир, 1976.
- 39.. Е.Ю. Шелковников, А.В. Тюриков, Б.А. Жуйков. Методика моделирования СТМ-изображений ультрадисперсных частиц с учетом взаимодействия поверхности и зондирующего острия // *Ползуновский альманах*. 2007. №3.
40. В.А. Быков, М.И. Лазарев, А.В. Тавров. Сканирующая зондовая микроскопия для науки и промышленности // *Компьютерра*. М.: ИД Компьютерра, 1997. №41.
41. E. Graugnard, T. Lee. TEM Pictures of STM Tips. Дата обновления: 19.07.2002. URL: <http://www.physics.purdue.edu/nanophys/uhvstm/tip.html> (дата обращения: 30.09.2010).
42. В.А. Карташев, В.В. Карташев. Исследование процесса электрохимической заточки иглы туннельного микроскопа // *Нано и микросистемная техника*. М.: Новые технологии, 2008. №4.
43. Е.Ю. Шелковников. Программно-аппаратные средства и алгоритмическая коррекция погрешностей измерений геометрических параметров наночастиц сканирующим туннельным микроскопом: Автореф. дис. докт. тех. наук. –

Ижевск, 2008.

44. Д.О. Филатов, А.В. Круглов. Исследование поверхности твердых тел методом сканирующей туннельной микроскопии (СТМ) // Н. Новгород: НОЦ СЗМ Нижегородского государственного университета, 2001.

45. В.А. Карташев, В.В. Карташев. Определение формы и размера острия иглы туннельного микроскопа. Нано и микросистемная техника. М.: Новые технологии, 2010. №10.

46. Э. Троелсен. С# и платформа .NET 3.0. СПб: Питер, 2008.

47. И. Грэхем. Объектно-ориентированные методы. Принципы и практика. М.: Изд. дом Вильямс, 2004.

48. А. Макки. Введение в .NET 4.0 и Visual Studio 2010 для профессионалов. М.: Изд. дом Вильямс, 2010.

49. Б. Страуструп. Язык программирования C++. М.: Бином, 1999.

50. Э. Таненбаум. Современные операционные системы. СПб: Питер, 2010.