

УСТРАНЕНИЕ ГРАДИЕНТА ОСВЕЩЕННОСТИ  
НА ПЭМ-ИЗОБРАЖЕНИЯХ. РЕАЛИЗАЦИЯ АЛГОРИТМА  
В ПРОГРАММЕ TEM IMAGING&ANALYSIS© 2021 г. А. В. Заводов<sup>1,\*</sup><sup>1</sup> *Всероссийский научно-исследовательский институт авиационных материалов, Москва, Россия*

\*E-mail: zavodovad@gmail.com

Поступила в редакцию 16.02.2021 г.

После доработки 16.02.2021 г.

Принята к публикации 24.02.2021 г.

Предложено решение широко распространенной проблемы ПЭМ-изображений – градиента освещенности. Устранение градиента освещенности осуществляется путем построчной обработки исходного изображения в 16-битном формате и его преобразованием в изображение с более узким диапазоном уровней серого [0–255]. Для устранения влияния неоднородностей изображения на результат обработки применяется полиномиальная аппроксимация исходных значений средней интенсивности  $I_{\text{mean}}$  строк и их ширины диапазона интенсивностей ( $I_{\text{max}} - I_{\text{min}}$ ). Разработанный алгоритм полностью устраняет градиент освещенности на изображениях, восстанавливая детали в затемненных и засвеченных участках, а также повышая контраст отдельных частей изображения. Приведен полный код алгоритма восстановления изображений для использования в программе TEM Imaging&Analysis.

DOI: 10.31857/S0023476121040238

## ВВЕДЕНИЕ

Получение качественного изображения в просвечивающем электронном микроскопе (ПЭМ) – трудоемкий и многосторонний процесс. Изображение должно иметь хороший контраст, отсутствие аберраций, размытия и искажений формы вследствие дрейфа образца, а также равномерную освещенность по всему полю зрения [1–4], которая напрямую связана с качеством и особенно с методу подготовки образцов для просвечивающей электронной микроскопии.

Идеальный образец (рис. 1) для просвечивающей электронной микроскопии можно описать как жесткое кольцо толщиной 50–100 мкм с тонкой, не более 100 нм, плоскопараллельной мембраной в центральной части. У образца должны быть некоторые ребра жесткости, позволяющие его транспортировать и не дающие тонкой части деформироваться. Современными методами подготовки, такими как струйная электролитическая полировка [5–7] и ионная полировка [8–11], едва ли удастся приблизиться к данному идеалу. Поэтому приходится работать с образцом переменной толщины. Как правило, именно угол так называемого “клина” определяет качество образца, доступную площадь изучения и возможности всего исследования.

Различная толщина образца приводит к появлению на изображении направленного градиента освещенности (рис. 2). Одна часть изображения засвечена, противоположная уходит в темноту. Эффект тем более выражен, чем быстрее изменяется толщина образца от края и чем большую область образца необходимо заснять. Единственным инструментом для улучшения изображения, доступным оператору микроскопа, является смещение электронного пучка от центра изображения в темную область. Однако такой метод подхо-

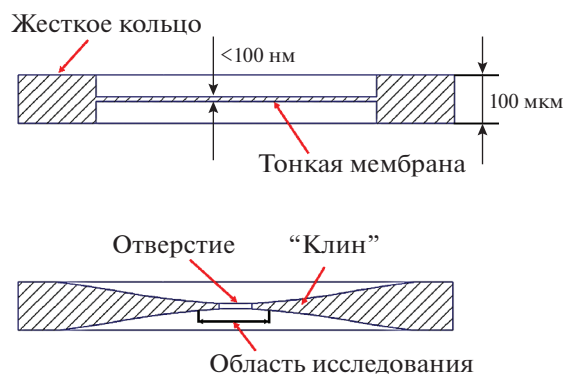


Рис. 1. Идеальный и реально получающийся образцы для просвечивающей электронной микроскопии.

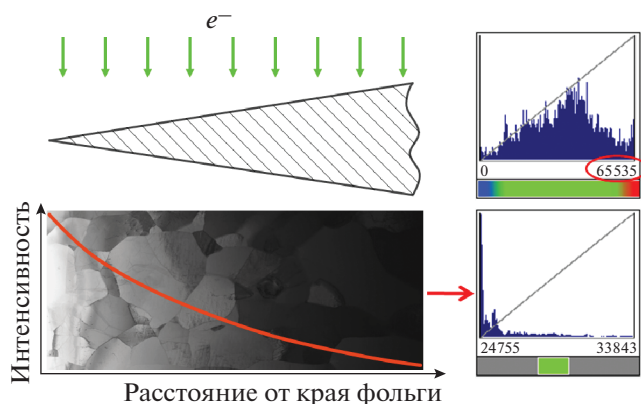


Рис. 2. Принцип образования градиента освещенности и получаемые гистограммы интенсивностей.

дит для относительно тонких образцов и небольших полей зрения. В большинстве случаев полностью устранить градиент освещенности не представляется возможным.

### ПОСТАНОВКА ЗАДАЧИ

В работе предлагается решение описанной выше проблемы, заключающееся в постобработке исходного файла полученного изображения. Для понимания сути задачи необходимо знать принцип отображения фотографий на мониторе, а также знать величину динамического диапазона детекторов, записывающих сигнал, формирующий изображение [12–14].

При съемке изображения детектор воспринимает пришедшие на него электроны и формирует сигнал, обрабатываемый управляющей схемой и программным обеспечением. Полученный сигнал записывается в относительных единицах, пропорциональных интенсивности сигнала или количеству электронов, пришедших на детектор. Диапазон воспринимаемых интенсивностей зависит от типа детектора, эта величина всегда значительно больше диапазона в 256 уровней серого, отображаемых на экране монитора. Так, максимальная ширина диапазона интенсивности, записываемая 16-битными значениями, составляет  $2^{16} = 65\,536$  уровней серого (рис. 2).

Для отображения исходного широкого диапазона на мониторе необходимо его преобразовать в 256 уровней серого. Это преобразование автоматически делают любые программы, работающие с изображениями широкого диапазона — «сырыми» файлами. При таком преобразовании неизбежна частичная потеря информации для наблюдателя. В случае равномерной освещенности преобразование исходного диапазона не несет ущерба итоговому снимку. Однако при наличии градиента освещенности все локальные различия

интенсивности, отвечающие за детали на изображении, попадают в узкие диапазоны в шкале [0–255], что помимо засветки и сильного затемнения приводит к низкому контрасту изображения. Гистограммы таких изображений сильно смещены в левую или правую часть в зависимости от того, преобладают темные области или засвеченные (рис. 2).

Поясним на примере. Если условно разделить изображение на четыре части по направлению градиента освещенности, то каждая часть будет отображена не во всем диапазоне серого [0–255], а лишь в его небольшой части (рис. 3). Так, самая темная четверть изображения будет отображаться на экране оттенками [0–35], что воспринимается глазом как темная область с низким контрастом и отсутствием деталей. Средние части изображения обладают большей шириной диапазона интенсивностей, но со значительным смещением влево. Гистограмма светлой части изображения обладает наиболее широким диапазоном и симметричной формой, что приводит к высокому контрасту при отображении на экране.

Каждая часть изображения в отдельности может быть более полноценно отображена во всем диапазоне [0–255] уровней серого. Совместив части изображения, каждая из которых преобразована во всем диапазоне серого, получаем изображение с искусственно расширенным диапазоном (HDR-изображение). Однако такое изображение (рис. 3) имеет существенный недостаток: видимость стыков отдельных частей изображения, обусловленная наличием градиента освещенности в каждой части. Таким образом, задача коррекции исходного изображения с градиентом освещенности сводится к представлению каждой отдельной строки (в случае вертикального градиента) в диапазоне [0–255] с последующим совмещением в одно изображение.

Существует большое количество методов восстановления изображений с широким диапазоном интенсивностей [15]. Настоящая работа является практической реализацией известных принципов построения HDR-изображений с учетом специфики ПЭМ-изображений. Все рассуждения и алгоритмы восстановления изображения приведены на примере работы в программе TEM Imaging&Analysis фирмы FEI. Однако общие принципы справедливы и для других программных продуктов, работающих с изображениями с широким диапазоном интенсивностей.

### РЕАЛИЗАЦИЯ В АЛГОРИТМАХ

*Преобразование по всей ширине диапазона каждой строки.* Суть алгоритма состоит в том, что в каждой отдельной строке исходного изображения весь диапазон интенсивностей разбивается на

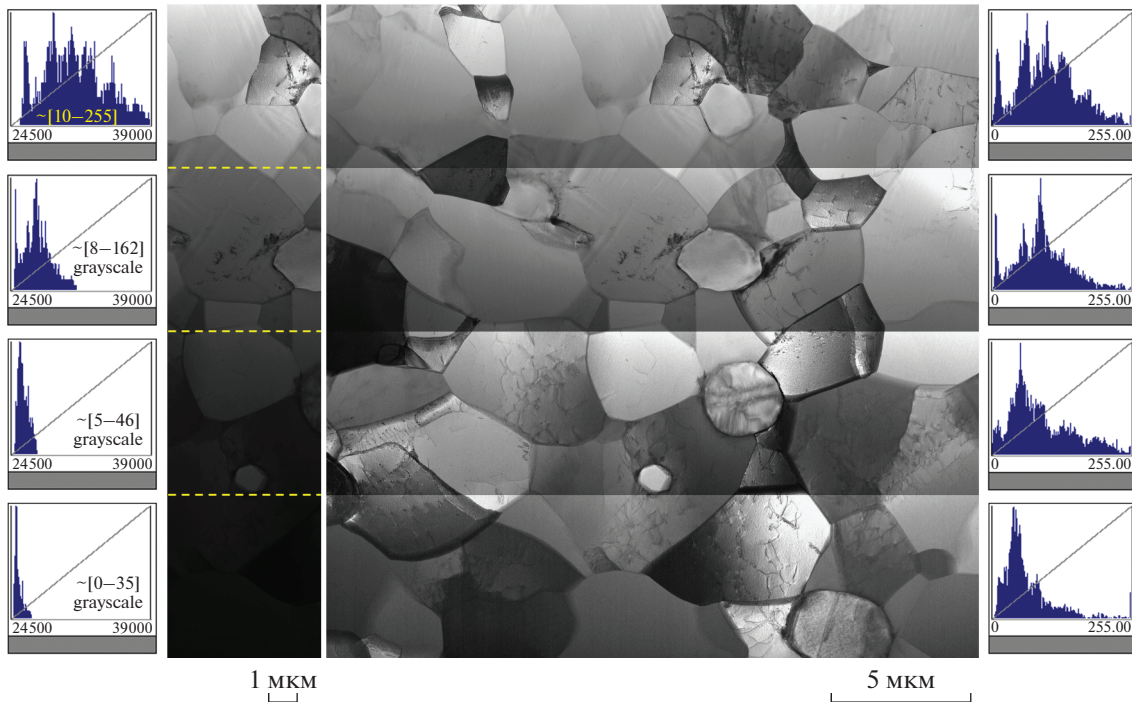


Рис. 3. Деление изображения вдоль направления градиента освещенности: гистограммы отдельных частей исходного изображения (слева) и после разделения и представления каждой части изображения в диапазоне [0–255].

256 интервалов. Далее значения интенсивностей, попадающие в один интервал, отождествляются с одним уровнем серого в диапазоне [0–255]. Преобразование интенсивности отдельного пикселя изображения можно записать как

$$I_i^* = \frac{I_i - I_{\min}}{\text{interval}}, \quad (1)$$

где  $\text{interval} = \frac{I_{\max} - I_{\min}}{256}$  – ширина интервала разбиения исходного диапазона интенсивностей в одной строке изображения;  $I_i, I_i^*$  – исходное и преобразованное значения интенсивности  $i$ -го пикселя строки изображения;  $I_{\min}, I_{\max}$  – исходные минимальное и максимальное значения интенсивности в отдельной строке изображения.

Типичный результат преобразования по такому алгоритму представлен на рис. 4. Как и ожидается, общий градиент устраняется, появляются ранее скрытые детали, контраст изображения максимальный. Однако все изображение состоит из полос различной яркости. Появление такого артефакта связано с тем, что исходные диапазоны интенсивностей различных строк сильно различаются из-за наличия в этих строках более темных или светлых областей.

Различия в исходных диапазонах интенсивностей соседних строк приводят к изменению ширины интервала разбиения, следовательно,

равные исходные значения интенсивностей в соседних строках пересчитываются на разные значения в диапазоне [0–255]. Это и создает эффект полосатого изображения. Если в строке присутствует более темный объект, то вся строка в целом будет более светлой. Если в строке яркое пятно, то в целом вся строка становится темнее.

Данный вариант алгоритма сильно зависит от крайних значений интенсивности  $I_{\min}$  и  $I_{\max}$  в каждой строке. Поэтому для полноценной работы алгоритма восстановления изображения эту зависимость необходимо устранить.

*Преобразование по ширине диапазона строк с аппроксимацией полиномом.* Настоящий алгоритм является усовершенствованием первого варианта алгоритма за счет замены значений  $I_{\min}$  и  $\text{interval}$ , взятых из исходных данных (рис. 5), на значения, лежащие на сглаживающей кривой.

Для работы этого алгоритма необходимы два набора значений, взятых из исходного изображения: среднее значение интенсивности  $I_{\text{mean}}$  пикселей в каждой строке и ширина диапазона интенсивности  $\text{range} = I_{\max} - I_{\min}$  в каждой строке, вычисляемая как разница между максимальным и минимальным значениями интенсивности в строке. Набор указанных выше значений аппроксимируется полиномами  $P_{\text{range}}(x)$  и  $P_{\text{mean}}(x)$   $n$ -ой степени вида  $P(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$ , ко-



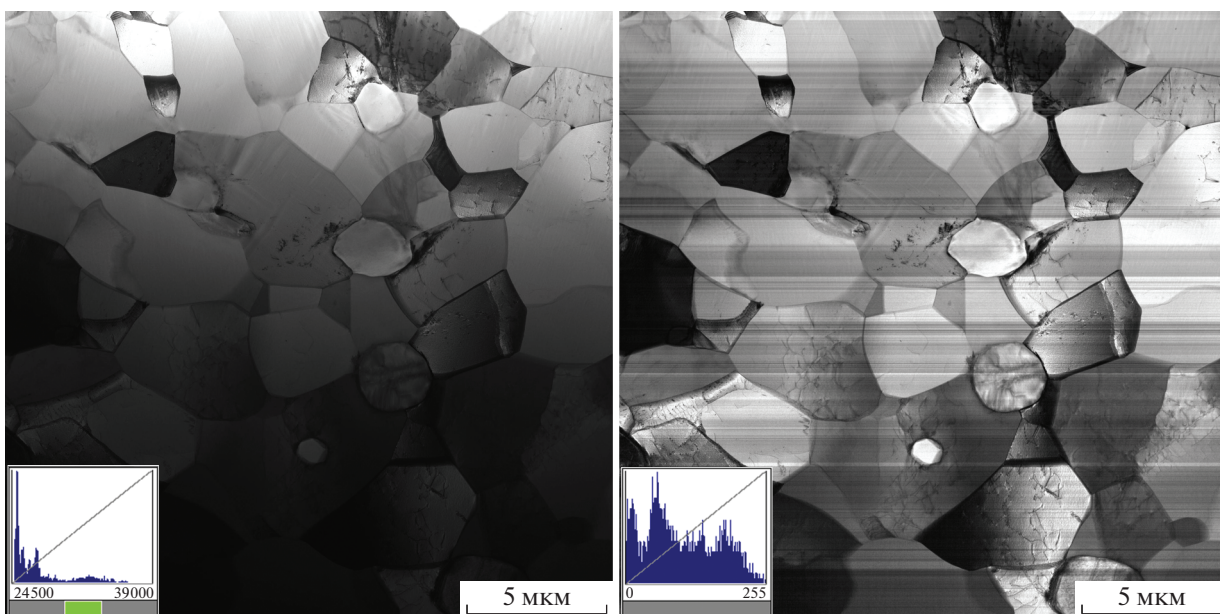


Рис. 4. Результат преобразования по первому алгоритму.

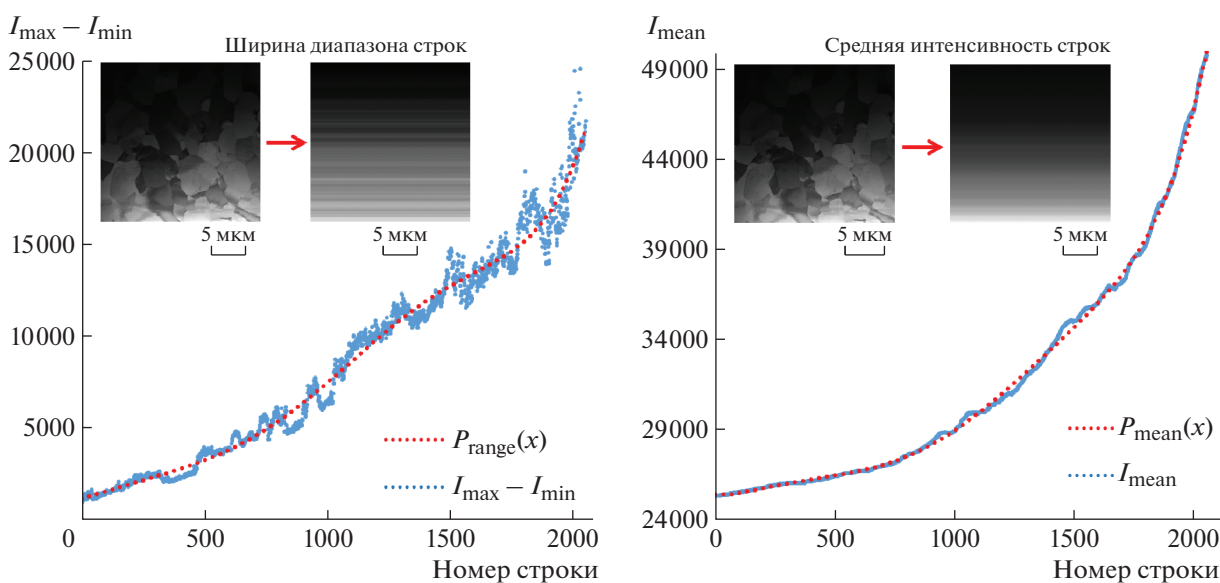


Рис. 5. Исходные значения ширины диапазонов интенсивностей  $I_{\max} - I_{\min}$  и средней интенсивности  $I_{\text{mean}}$  каждой строки и аппроксимирующие кривые.

торые используются для расчета значений ширины интервала  $\text{interval}$  и минимального значения интенсивности  $I_{\min}$  в формуле (1). В данном случае независимой переменной полиномов является номер строки изображения, иначе говоря, набор натуральных чисел от 0 до 2047 (в зависимости от разрешения снимка).

Отметим, что  $I_{\min}$  (рис. 6) здесь вычисляется не как значение полинома, аппроксимирующего набор исходных минимальных значений интенсивностей в строках, а по следующей формуле:

$$I_{\min} = I_{\text{mean}} - \frac{\text{range}}{2} = P_{\text{mean}}(x) - \frac{P_{\text{range}}(x)}{2}, \quad (2)$$



где  $x$  – номер строки изображения. При таком вычислении значения  $I_{\min}$  отражают изменения как средней интенсивности, так и ширины диапазона в строке, что позволяет устранить влияние темных частиц в строках изображения и осуществить преобразование в диапазон  $[0-255]$  более симметрично относительно средних значений интенсивности.

Алгоритм можно представить в виде следующей последовательности шагов.

1. Формирование набора пар значений  $[N\text{ строки}, I_{\text{mean}}]$  вдоль направления градиента освещенности.

2. Формирование набора пар значений  $[N\text{ строки}, I_{\text{max}} - I_{\text{min}}]$  вдоль направления градиента освещенности.

3. Вычисление коэффициентов  $a_i$  аппроксимирующих полиномов  $P_{\text{range}}(x)$  и  $P_{\text{mean}}(x)$  для обоих наборов пар значений (метод наименьших квадратов).

4. Вычисление ширины диапазона интенсивностей конкретной строки путем подстановки номера строки в аппроксимирующий полином ( $\text{range} = P_{\text{range}}(N\text{ строки})$ ).

5. Вычисление ширины интервала по формуле:  

$$\text{interval} = \frac{\text{range}}{256}.$$

6. Вычисление среднего значения интенсивности строки путем подстановки номера строки в аппроксимирующий полином ( $I_{\text{mean}} = P_{\text{mean}}(N\text{ строки})$ ).

7. Вычисление минимального значения интенсивности в строке по формуле (2).

8. Преобразование исходной интенсивности в значения в диапазоне  $[0-255]$  по формуле (1).

Шаги 4–8 повторяются для каждой строки изображения. Реализация алгоритма на языке JavaScript представлена в Приложении 1.

## РЕЗУЛЬТАТЫ

Результат работы итогового алгоритма показан на нескольких примерах (рис. 7) с различной плавностью градиента освещенности. Алгоритм справляется с восстановлением даже тех изображений, на которых одновременно присутствуют засвеченная область и широкая область с полным затенением без видимых деталей (рис. 7а, 7б). Устранение засвеченной области и выявление в ней деталей возможны благодаря тому, что в исходном файле изображения пиксели засвеченной области имеют различные значения, не достигающие максимальных, т.е. информация о деталях изначально сохранена.

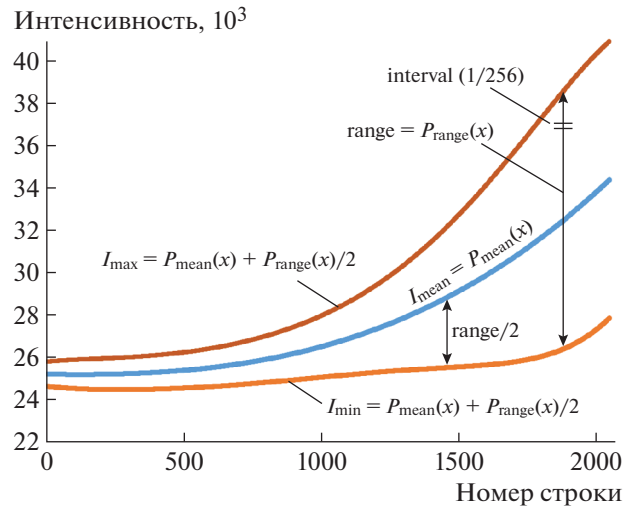
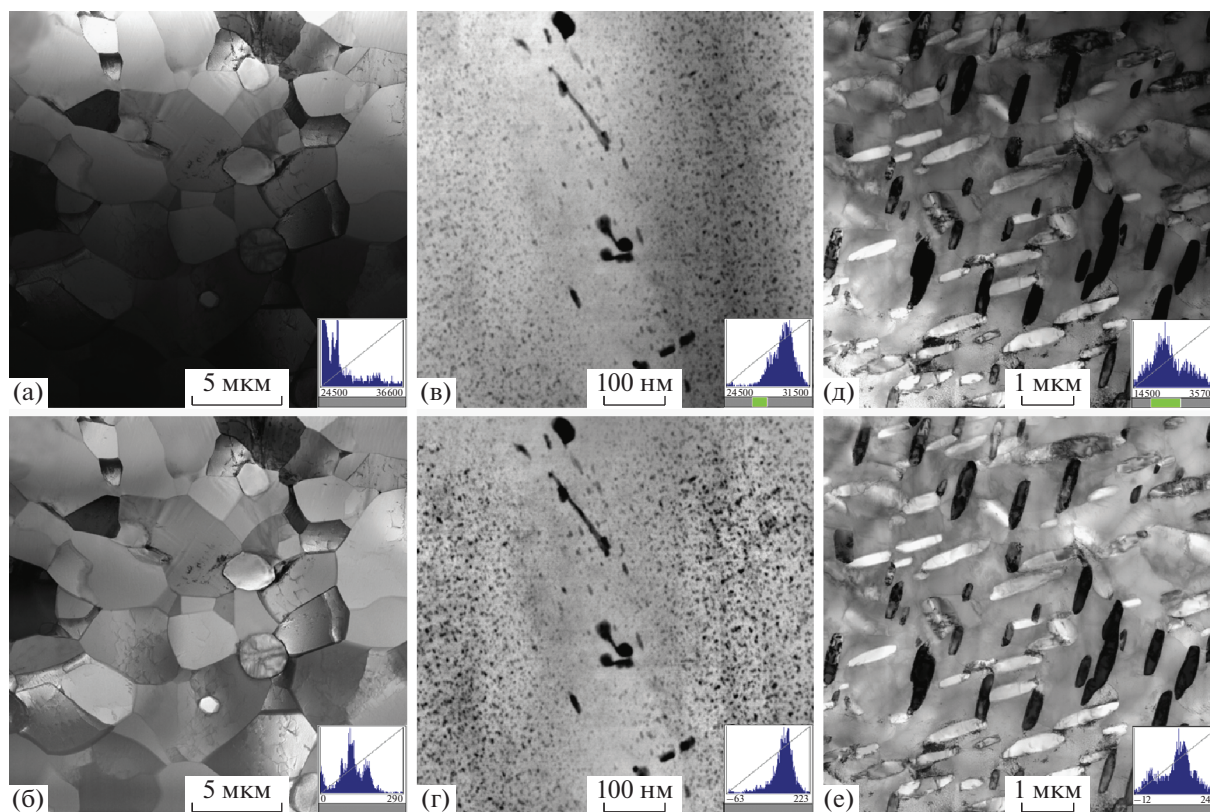


Рис. 6. Графическое представление значений, используемых при вычислениях.

Также отсутствуют полосы на изображении в отличие от первого алгоритма, когда в диапазон  $[0-255]$  преобразовывается исходный диапазон по всей ширине. В отличие от первого варианта алгоритма ширина диапазона получаемого изображения больше 256 уровней, так как крайние значения исходного диапазона могут выходить за границы  $[0-255]$ . Типичная ширина диапазона преобразованного изображения составляет от 300 до 500 единиц. Границы диапазона при этом уходят как в отрицательную область, так и в значения выше 255, например полный диапазон интенсивностей на рис. 7б составляет  $[-53-442]$ . Это приводит к тому, что некоторые детали в очень темных участках при отображении на экране могут терять контраст или вовсе пропадать. Однако это не является значительной проблемой, так как глаз плохо воспринимает слабые различия в темных участках, следовательно, ими можно пренебречь.

Обратим внимание на изменение формы гистограмм распределения интенсивностей. Максимум гистограмм после обработки смещается в центральную часть, форма распределения становится более симметричной, что является показателем правильно экспонированного кадра и равномерной освещенности. При обработке изображения также повышается контрастность отдельных участков изображения без переэкспонирования других областей (рис. 7в, 7г), чего невозможно сделать стандартными методами коррекции, такими как изменение яркости, контрастности или  $\gamma$ -коррекцией.

Применение этого алгоритма не ограничивается изображениями только с вертикальным или



**Рис. 7.** Результаты работы итогового алгоритма устранения градиента освещенности: а, б – вертикальный градиент с сильным перепадом, в, г – горизонтальный градиент с восстановлением контраста отдельных областей, д, е – восстановление диагонального градиента.

только горизонтальным градиентом освещенности. На рис. 7д, 7е показан пример восстановления изображения с диагональным (сверху справа – вниз налево) градиентом освещенности. Чтобы восстановить такое изображение, необходимо дважды применить этот алгоритм к исходному изображению, повернув его на  $90^\circ$  после первого прохода. При этом на втором проходе алгоритм работает с ужатым диапазоном интенсивностей, что не мешает ему устранять градиент за счет нормализации яркости отдельных строк.

*Выбор степени полинома, ошибка перехода через ноль.* Код алгоритма преобразования изображения составлен так, что пользователь может быстро поменять степень аппроксимирующего полинома, переписав в коде скрипта одно значение (Приложение 1, строка 137). В большинстве случаев для корректной обработки изображения достаточно использовать третью степень полинома. Чем большая степень полинома используется, тем точнее описываются исходные наборы значений-показателей градиента, но при этом возрастает вероятность получения излишнего контраста.

Однако в ряде случаев график аппроксимирующего полинома может переходить через 0, что

негативно сказывается на результате обработки изображения (рис. 8). Происходит это в тех случаях, когда строки изображения имеют очень узкую ширину диапазона интенсивностей, что на изображении выглядит как однородный фон. В таком случае  $P_{\text{range}}(x)$  принимает околонулевые значения, и на изображении появляется полоса, по разные стороны от которой наблюдаются очень высокие положительные и отрицательные значения интенсивностей, порядка  $\pm 10^5$  и более.

Для устранения этого эффекта необходимо изменить степень аппроксимирующего полинома на любую другую, где такого эффекта не будет наблюдаться. Неправильно подобранная степень полинома может отражаться в виде засвеченных или затемненных полос, как правило, вблизи краев изображения. Эта проблема также решается изменением степени аппроксимирующего полинома в коде алгоритма.

## ВЫВОДЫ

В работе предложено эффективное решение распространенной проблемы ПЭМ-изображений – градиента освещенности. Алгоритм вос-

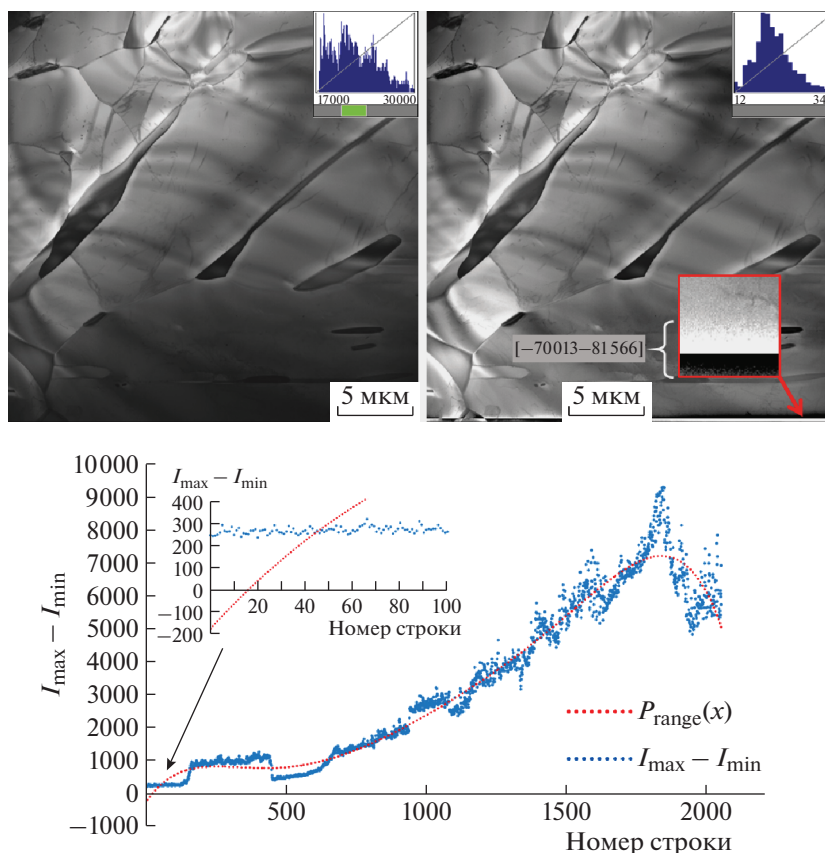


Рис. 8. Артефакты на результирующем изображении, обусловленные околонулевыми значениями аппроксимирующего полинома.

становления позволяет полностью устранить неравномерность освещения изображения в любых направлениях, восстанавливая детали изображения как в темных областях, так и условно засвеченных. Следствием работы алгоритма становится повышение контраста отдельных частей изображения, без переэконтрастирования изображения в других областях. Результат такого преобразования не может быть достигнут стандартными методами коррекции, основанными на работе с гистограммой распределения интенсивности.

Основная идея алгоритма восстановления изображения лежит в максимально эффективном использовании всего исходного 16-битного диапазона интенсивностей изображения путем построчного преобразования в диапазон  $[0-255]$  уровней серого. Для реализации предложенного алгоритма приведены все необходимые фрагменты кода скрипта для программы TEM Imaging&Analysis (Приложение 1). Наилучшие результаты обработки изображений достигаются при использовании исходных файлов типа .emf с широким диапазоном интенсивностей. Алгоритм успешно справляется и с обработкой сжатых типов изображения, таких как .bmp и .tiff.

## ПРИЛОЖЕНИЕ 1

Представлена реализация алгоритма преобразования изображений с целью устранения градиента освещенности в программе TEM Imaging&Analysis (TIA) фирмы FEI. Для того чтобы использовать ниже приведенный код, необходимо его вставить в пример открытого скрипт-файла, располагающегося в папке `C:\Program-Files\FEI\TIA\Components\Demo\ProcessingDemo`. Имя функции должно совпадать с объявленным в начале текста скрипта. Далее установить измененный файл скрипта в рабочем окне программы. Коррекция изображения с разрешением  $2048 \times 2048$  занимает не более 1 с.

При написании кода скрипта для TIA необходимо помнить некоторые особенности исполняющей среды:

- TIA воспринимает скрипты, написанные на языке JavaScript, стандарта не выше ES3 (1999 г). Таким образом, код скрипта должен быть написан без применения упрощенного синтаксиса;
- при работе с изображением необходимо предварительно создать его копию (строка 126)



и все изменения проводить с ней. Работа с исходным изображением не приведет к его изменению;

– при вычислениях необходимо помнить, что в ТИА используется комплексный тип чисел

(ComplexNumber). При операциях с вещественными числами необходимо предварительно их преобразовать в тип ComplexNumber.

Описание всех используемых объектов, типов и методов представлено в справке ТИА.

```

121 // ProcessingDemo_Degradient
122 function ProcessingDemo_Degradient ()
123 {
124     var Image = this.SelectedImage();
125     if (Image != null) {
126         TempImage = Image.Data; //создание в памяти копии обрабатываемого изображения
127         var Ysize = Image.Data.PixelsY();
128
129         var masint = new Array(Ysize); //создание одномерного массива средних интенсивностей строк
130         for(var i = 0; i < Ysize; i++)
131             {masint[i] = esMath.Mean(TempImage.LineAtY(i)).Real}
132
133         var masrg = new Array(Ysize); //создание одномерного массива диапазонов интенсивностей строк
134         for(var i = 0; i < Ysize; i++)
135             {masrg[i] = esMath.Sub(esMath.Max(TempImage.LineAtY(i)), esMath.Min(TempImage.LineAtY(i))).Real}
136
137         var power = 6; //степень аппроксимирующего полинома
138         var CoefMatr = BuildCoefMatrix(Ysize,power); //квадратная матрица коэффициентов
139         var CoefMatr2 = BuildCoefMatrix(Ysize,power);
140
141         var Matrrg = FullMatrix(CoefMatr,masrg); //добавление столбца свободных членов
142         var Matrrg = PolynomialRegression(Matrrg); //вычисление коэффициентов полинома
143
144         var Matrint = FullMatrix(CoefMatr2,masint); // то же для средних интенсивностей строк
145         var xint = PolynomialRegression(Matrint);
146
147         for(var i = 0; i < Ysize; i++) { //основной блок преобразования шаги 4 - 8
148             var range = Estimate(xrg,i);
149             var interval = Esv.ComplexNumber((range/256),0);
150             var min = Esv.ComplexNumber((Estimate(xint,i)-range/2),0);
151             TempImage.LineAtY(i) = esMath.div(esMath.sub(TempImage.LineAtY(i),min), interval); //формула (1)
152         }
153         Image.Data = TempImage; //запись изменённого изображения
154         Image.DisplayIntensityRange = Image.IntensityRange(0.99);
155     }
156     else
157         Esv.WarningDialog(this.ErrorMessage);
158 }

```

Рис. 1. Основной блок вычислений с комментариями. Конечный вариант алгоритма.

```

53 //Создание квадратной матрицы коэффициентов
54 function BuildCoefMatrix(Ysize,power)
55 {
56     //массив сумм степеней чисел от 0 до Ysize
57     var arrx = [Ysize];
58     for (var i=1; i<=2*power; i++) {
59         var sum = 0;
60         for (var j = 1; j<Ysize; j++) {
61             sum = sum+Math.pow(j,i);
62         }
63         arrx.push(sum);
64     }
65     //заполнение квадратной матрицы
66     var Mcoef = new Array(power+1);
67     for (var i=0; i<=power; i++) {
68         Mcoef[i] = [];
69         for (var j=0; j<=power; j++) {
70             Mcoef[i].push(arrx[j+i]);
71         }
72     }
73     return Mcoef;
74 }
75
76 //Добавление столбца свободных членов
77 function FullMatrix(M,arr)
78 {
79     for (var i=0; i<M.length; i++) {
80         var sum = 0;
81         for (var j=0; j<arr.length; j++) {
82             sum = sum+Math.pow(j,i)*arr[j];
83         }
84         M[i].push(sum);
85     }
86     return M;
87 }

```

Рис. 2. Составление матрицы для решения системы линейных уравнений.

```

89 //Нахождение коэффициентов полинома степени n
90 function PolynomialRegression(Matrix)
91 { //приведение матрицы к треугольному виду
92     var n = Matrix.length;
93     for (var i=0; i<n-1; i++) {
94         for (var j=i+1; j<n; j++) {
95             var c = Matrix[j][i]/Matrix[i][i];
96             for (var k=i+1; k<n+1; k++) {
97                 Matrix[j][k] = Matrix[j][k] - c*Matrix[i][k];
98             }
99         }
100     }
101     //обратный ход
102     var x = [];
103     for (var i=n-1; i>=0; i--) {
104         var SolvedTerms = 0;
105         for (var j=0; j<x.length; j++) {
106             SolvedTerms = SolvedTerms+x[j]*Matrix[i][n-1-j];
107         }
108         x.push((Matrix[i][n] - SolvedTerms)/Matrix[i][i]);
109     }
110     return x.reverse();
111 }
112
113 //вычисление значения полинома в точке x
114 function Estimate(arr,x)
115 {
116     var value = 0;
117     for (var i=0; i<arr.length; i++) {
118         value = value + arr[i]*Math.pow(x,i);
119     }
120     return value;
121 }

```

Рис. 3. Решение системы линейных уравнений методом Гаусса и вычисление значений аппроксимирующего полинома.

## СПИСОК ЛИТЕРАТУРЫ

1. *Williams D.B., Barry C.C.* Transmission electron microscopy A Textbook for Materials Science. New York, Springer Science+Business Media. 2009. 775 p.
2. *Фулъц Б., Хау Дж.М.* Просвечивающая электронная микроскопия и дифрактометрия материалов. М.: Техносфера, 2011. 904 с.
3. *Horsssen E.P., Janssen B.J., Kumar A. et al.* // IFAC J. Systems Control. 2020. V. 11. Art. 100074. <https://doi.org/10.1016/j.ifacsc.2020.100074>
4. *Jin P., Li X.* // J. Microscopy. 2015. V. 260. № 3. P. 268. <https://doi.org/10.1111/jmi.12293>
5. *Horvath B., Schaublin R., Dai Y.* // Nucl. Instrum. Methods Phys. Res. B. 2019. V. 449. P. 29. <https://doi.org/10.1016/j.nimb.2019.04.047>
6. *Necip Ü.* // Mater. Charact. 2008. V. 59. P. 547. <https://doi.org/10.1016/j.matchar.2007.04.003>
7. *Zieliński W., Płociński T., Kurzydłowski K.J.* // Mater. Charact. 2015. V. 104. P. 42. <https://doi.org/10.1016/j.matchar.2015.04.003>
8. *Chen L., Wang Y., Zhang Z.* // Micron. 2018. V. 2018. P. 46. <https://doi.org/10.1016/j.micron.2018.04.007>
9. *Richter J., Chrapoński J.* // Int. J. Refract. Met. Hard Mater. 2018. V. 75. P. 163. <https://doi.org/10.1016/j.ijrmhm.2018.04.016>
10. *Lotnyk A., Poppitz D., Ross U. et al.* // Microelectron. Reliab. 2015. V. 55. P. 2119. <https://doi.org/10.1016/j.microrel.2015.07.005>
11. *Thangadurai P., Lumelsky Y., Silverstein M.S., Kaplan W.D.* // Mater. Charact. 2008. V. 58. P. 1623. <https://doi.org/10.1016/j.matchar.2008.02.007>
12. *Faruqi A.R., McMullan G.* // Nucl. Instrum. Methods Phys. Res. A. 2018. V. 878. P. 180. <https://doi.org/10.1016/j.nima.2017.07.037>
13. *Tate M.W.* // Microsc. Microanal. 2016. V. 22. № 1. P. 237. <https://doi.org/10.1017/S1431927615015664>
14. *Halazonetis D.J.* // Am. J. Orthod. Dentofacial Orthop. 2005. V. 127. № 5. P. 387. <https://doi.org/10.1016/j.ajodo.2004.07.025>
15. *Reinhard E., Heidrich W., Debevec P. et al.* High dynamic range imaging: acquisition, display, and image-based lighting. Morgan Kaufmann. 2010. 672 p.