

МЕТОДИКА ОБРАБОТКИ ТОПОГРАФИЧЕСКИХ СТМ-ИЗОБРАЖЕНИЙ ДЛЯ ИХ ИДЕНТИФИКАЦИИ С ПРИМЕНЕНИЕМ НЕЙРОННОЙ СЕТИ

Суворов А.С.

Институт прикладной механики УрО РАН
г. Ижевск

Исследования ультрадисперсных частиц кластерных материалов (УДЧ КМ) с применением сканирующего туннельного микроскопа (СТМ) являются важным этапом при создании новых перспективных материалов на их основе. Поскольку СТМ позволяет получать топографические изображения поверхности постоянного туннельного тока (величина которого в каждой точке над поверхностью определяется суперпозицией определенных электронных состояний, а, следовательно, и квантово-электронным строением исследуемой УДЧ и подложки) то идентификация подобных СТМ-изображений достаточно затруднена и требует привлечения достоверных изображений, построенных теоретическим путем. Однако, даже при наличии достоверных теоретических изображений кластерных частиц интерпретация экспериментальных данных может стать непосильной задачей, поскольку «зашумленность» изображения (неизбежная в силу постоянного наличия помех различной природы в процессе сканирования) не позволяет человеческому разуму правильно интерпретировать изображение. Именно поэтому создание методик интерпретации экспериментальных СТМ-изображений является в настоящее время важной и актуальной задачей.

Изображения, получаемые с помощью СТМ, обладают свойственными аналоговым сигналам недостатками: отсутствием строго ограниченной шкалы координат, произвольным углом отображения частиц и присутствием шумов и искажающих факторов. Для исключения влияния данных недостатков предполагается использовать следующие меры, которые можно представить в виде алгоритма:

- выделение участка изображения с объемной частицей на подложке графита;
- нахождение характерных для этой частицы профилограмм;
- нормировка профилограмм в заданном диапазоне.

Для построения изображения используются результаты, полученные с помощью па-

ПОЛЗУНОВСКИЙ АЛЬМАНАХ №2 2009

кета квантово-химических расчетов GAMESS. Важнейшей характеристикой данных для построения изображений являются не столько размеры и точные координаты каждой точки изображения, сколько соотношения высот, по которым строится объемное изображение.

Для обнаружения частицы на изображении используется аппарат нейронных сетей с использованием методики обучения адаптивного усиления (AdaBoost) и применение примитивов (Haar-like features) для описания объекта.

Процедура распознавания объекта классифицирует объект частицы на изображении, основываясь на значениях примитивов. Анализ изображения с помощью примитивов, а не анализ точек непосредственно, позволяет существенно сократить вычислительные затраты и количество наборов изображений, используемых для обучения.

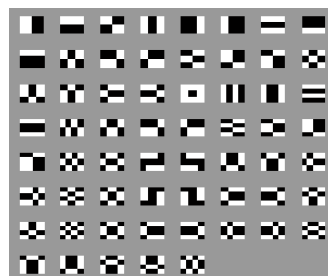


Рисунок 1 – Графические примитивы, применяемые для классификации объектов

Значением каждого примитива является разница суммы значений точек в белой и черной областях:

$$V = \sum_w I(x, y) - \sum_b I(x, y) . \quad (1)$$

При размере изображения 24x24 полный набор классификатора состоит из более, чем 180 тыс. примитивов. Для снижения временных затрат вычисления значений каждого примитива из набора используется метод интегрирования изображения – для каждой точки производится суммирование значений точек в области, лежащей слева и сверху данной точки:

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y'), \quad (2)$$

где $ii(x, y)$ – интегрированное изображение; $i(x, y)$ – оригинальное изображение. С помощью вычислений изображение интегрируется за одну итерацию:

$$s(x, y) = s(x, y-1) + i(x, y) \quad (3)$$

$$ii(x, y) = ii(x-1, y) + s(x, y) \quad (4)$$

Таким образом, подсчет значения каждого примитива сводится к вычислению разницы между значением правой-нижней и левой-верхней точек белой и черной области соответственно (рисунок 2), т.е.:

$$\begin{aligned} i_2 &= B + A \\ i_3 &= A + C \end{aligned} \quad (5)$$

$$i_4 = A + B + C + D$$

$$i_D = i_4 + i_1 - (i_2 + i_3),$$

где A – сумма точек слева-сверху от точки 1; B – сумма точек слева-сверху от точки 2; C – сумма точек слева-сверху от точки 3; D – сумма точек слева-сверху от точки 4.

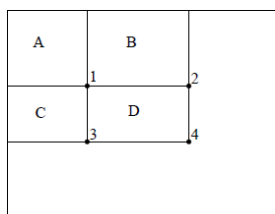


Рисунок 2 – Интегрированное изображение

Даже используя интегрированное изображение, применение каждого примитива из большого набора требует больших вычислительных затрат. Снизить их позволяет использование алгоритма AdaBoost.

AdaBoost (сокр. Adaptive Boosting) является адаптивным алгоритмом обучения – каждый следующий классификатор строится по объектам, неверно классифицированным предыдущими классификаторами. AdaBoost чувствителен к шуму в данных и «выбросам». Однако он менее подвержен переобучению, чем многие другие подобные алгоритмы. Применяется для обнаружения слабых аномалий на фоне сильных помех в данных, которые не связаны с изображениями и имеют другую природу.

AdaBoost вызывает классификатор в цикле $t = 1, \dots, T$. После каждого вызова обновляется распределение весов D_t , которые отвечают важности каждого из объектов обучающего множества для классификации. На каждой итерации веса каждого неверно клас-

сифицированного объекта возрастают (или аналогично, вес каждого корректно классифицированного объекта уменьшается); таким образом, новый классификатор «фокусируется» на этих объектах. Алгоритм является эффективным и быстрым для задач распознавания образов, так как из множества простых классификаторов (не требующих больших вычислительных затрат) в процессе обучения составляется комбинация, с достаточной степенью точности определяющая наличие и положение объекта на изображении.

Для обучения классификаторов используются 2 набора изображений: положительные (изображения объекта) и отрицательные (фон) (рисунок 3).

Рассмотрим алгоритм для задачи построения бинарного классификатора.

Дано: $(x_1, y_1), \dots, (x_m, y_m)$, где $x_i \in X, y_i \in Y = \{-1, +1\}$.

Инициализация $D_1(i) = \frac{1}{m}, i = 1, \dots, m$, для каждого $t = 1, \dots, T$:

- Определение классификатора $h_t : X \rightarrow \{-1, +1\}$, который минимизирует взвешенную ошибку классификации: $h_t = \arg \min_{h_t \in H} \varepsilon_j$, где

$$\varepsilon_j = \sum_{i=1}^m D_t(i) [y_i \neq h_j(x_i)].$$

- Если величина $\varepsilon_t \geq 0.5$, то «конец».

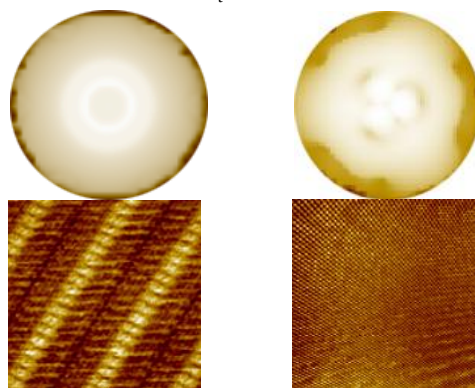


Рисунок 3 – Положительные и отрицательные изображения

- Выбор $a_t \in R$ (обычно $a_t = \frac{1}{2} \ln \frac{1-\varepsilon_t}{\varepsilon_t}$, где ε_t взвешенная ошибка классификатора h_t).
- Обновление:

$$D_{t+1}(i) = \frac{D_t(i)e^{-\alpha_t y_i h_t(x_i)}}{Z_t}, \text{ где } Z_t \text{ является}$$
 нормализующим параметром (выбранным так, чтобы D_{t+1} являлось распределением вероятностей, то есть $\sum_{i=1}^m D_{t+1}(i) = 1$).

Построение результирующего классификатора:

$$H(x) = \text{sign}\left(\sum_{t=1}^T \alpha_t h_t(x)\right).$$

Выражение для обновления распределения D_t должно быть сконструировано таким образом, чтобы выполнялось условие:

$$e^{-\alpha_t y_i h_t(x_i)} = \begin{cases} < 1, & y(i) = h_t(x_i) \\ > 1, & y(i) \neq h_t(x_i) \end{cases}$$

Таким образом, после выбора оптимального классификатора h_t для распределения D_t , объекты x_t (которые классификатор h_t идентифицирует корректно) имеют веса меньшие, чем те, которые идентифицируются некорректно. Следовательно, когда алгоритм тестирует классификаторы на распределении D_{t+1} , он будет выбирать классификатор, который лучше идентифицирует объекты, неверно распознаваемые предыдущим классификатором.

Обученная таким образом нейронная сеть далее принимает форму каскада (рис. 4): если первый слой квалифицирует изображение как положительное, сеть начинает обрабатывать второй слой; в случае, если изображение квалифицировано как отрицательное, сеть прекращает работу.

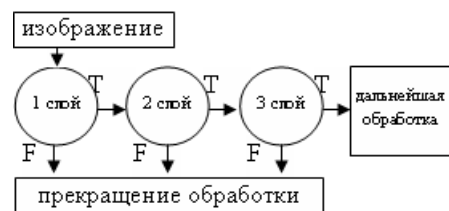


Рисунок 4 – Обработка изображения каскадом

Каждый последующий слой обладает большей степенью точности обнаружения объекта, но требует больших вычислительных затрат. Таким образом, существенно снижается время обработки изображения.

Использование каскада помогает на ранних этапах выделить изображения, со-

держащие объект, или прекратить обработку, при его отсутствии. Для найденных частиц (рисунок 5) проводится ряд профилограмм.

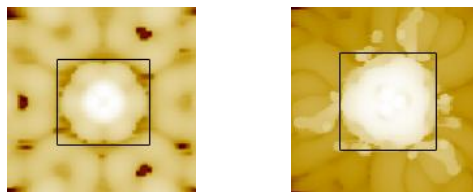


Рисунок 5 – Результаты распознавания изображений

На полученных профилограммах выделяется участок, соответствующий частице, и производится нормировка по длине и ширине в диапазоне 0..1 (рис. 6). Таким образом, в ходе обработки трехмерного изображения рядом общих и узкоспециализированных алгоритмов, формируется набор нормализованных профилей, пригодных для автоматического распознавания с помощью аппарата нейронных сетей.

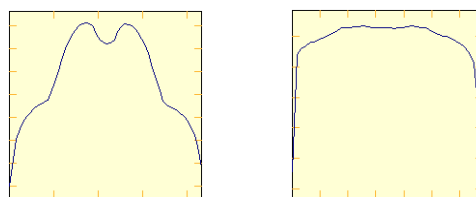


Рисунок 6 – Результат нормализации профилей частицы

Обработка изображения первым слоем, состоящим из классификатора с двумя примитивами, требует 60 операций микропроцессора, в то время, как однослойная нейронная сеть производит обработку изображения в 20 раз медленнее. Это позволяет говорить об оптимальном использовании алгоритма обучения AdaBoost и Хаг-примитивов в задачах распознавания объекта на изображениях.

СПИСОК ЛИТЕРАТУРЫ

1. P. Viola, M. Jones. Rapid Object Detection using a Boosted Cascade of Simple Features // Accepted conference on computer vision and pattern recognition.- Cambridge, 2001.- p.1-9.
2. R. Lienhart, A. Kuranov, V. Pisarevsky. Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection // Microprocessor Research Lab, Intel Labs.-USA, 2002.-c.1-8.
3. J. Young, E. Sharlin, J. Boyd. Implementing Bubblegrams: The Use of Haar-Like Features for Human-Robot Interaction // Accepted conference on computer vision and pattern recognition.- University of Calgary, 2002.-p.1-6.c.