

УДК 004.932

ОБРАБОТКА ИЗОБРАЖЕНИЙ ДЛЯ ОПРЕДЕЛЕНИЯ ПРОЧНОСТНЫХ ПАРАМЕТРОВ ПОЛИМЕРНЫХ ПЛЕНОК

С.И. РОГОВСКИЙ, О.В. ТАНАНА, Н.В. ВАБИЩЕВИЧ,
канд. физ.-мат. наук, доц. С.А. ВАБИЩЕВИЧ
(Полоцкий государственный университет)

Рассмотрены вопросы использования цифровой обработки изображений поверхности полимера при испытаниях на микротвердость для определения геометрических размеров отпечатков, зоны разрушения и деформации. Построен алгоритм обработки изображения и реализована программа обработки изображений. Результаты моделирования могут быть использованы при испытаниях полимерных пленок на микротвердость для определения прочностных характеристик: микротвердости, трещиностойкости, удельной энергии отслаивания пленки.

Ключевые слова: обработка изображения, программирование, поверхность, микроиндентирование, прочностные характеристики.

Введение. Информационные технологии широко распространены во всех сферах жизнедеятельности человека. Они дают возможность быстро и наиболее точно проводить расчеты, обрабатывать информацию и исключить субъективность в анализе результатов. Однако при организации научно-исследовательской работы далеко не всегда достаточно возможностей стандартных пакетов программных продуктов для анализа графической информации. Кроме того, актуальность исследований в области цифровой информации постоянно возрастает, что связано с высокой информативностью и доступностью ее получения. Все это свидетельствует о необходимости создания специализированных программных продуктов, позволяющих обрабатывать специфическую информацию, полученную в виде фотографии или видеоснимков.

Целью настоящей работы является разработка программы, позволяющей получить информацию о прочностных характеристиках материалов на основании анализа отпечатков, образующихся при микроиндентировании образцов.

Физические аспекты проекта. Современный процесс производства полупроводниковых приборов предусматривает ряд технологических обработок, в результате которых, помимо прочих, изменяются механические свойства материалов, что, в свою очередь, может приводить к появлению микротрещин, царапин, сколов и иных дефектов поверхности. Указанные явления и процессы приводят к повышению процента выбраковки готовой продукции и снижают экономическую эффективность производства приборов. Установлено, что механические свойства полупроводников зависят от изменения их дефектно-примесного состава, который существенно изменяется при технологических обработках. С целью оптимизации условий и режимов проведения указанных обработок необходимо проводить постоянный контроль прочностных характеристик материалов на каждом этапе производства и разрабатывать научно обоснованные методики изготовления полупроводниковых структур, которые учитывали бы изменение механических свойств используемого материала.

Для определения прочностных характеристик твердых материалов используются различные методы, однако методом, наиболее приближенным к реальному технологическому процессу, является микроиндентирование, т.к. оно позволяет наиболее точно моделировать контактное взаимодействие абразивных частиц с обрабатываемым материалом. Измерения микротвердости проводятся по стандартной методике [1]. При этом на испытываемую поверхность образца наносятся 30–50 отпечатков под действием статической нагрузки, приложенной к индентору. По измеренным диагоналям отпечатков и длине образовавшихся микротрещин определяют микротвердость, микрохрупкость, трещиностойкость материала, а также удельную энергию отслаивания пленки.

Размеры отпечатков и трещин в современных условиях можно получать как в результате проведения непосредственных экспериментов по микроиндентированию, так и на основании анализа цифровых фотографий изображений отпечатков. Однако и в том и в другом случае необходимо учитывать ряд условий, снижающих субъективность в проведении измерений. Так, например, практически невозможно точно измерить длину трещин от индентора, которые имеют сложную форму. Оценка балла микрохрупкости также может носить субъективный характер. Повышение эффективности проведения указанных выше измерений и расчетов возможно в результате их автоматизации с применением информационных технологий. В частности, существует необходимость создания программного продукта для распознавания цифровых изображений, позволяющего проводить на основании анализа фотоснимков полный

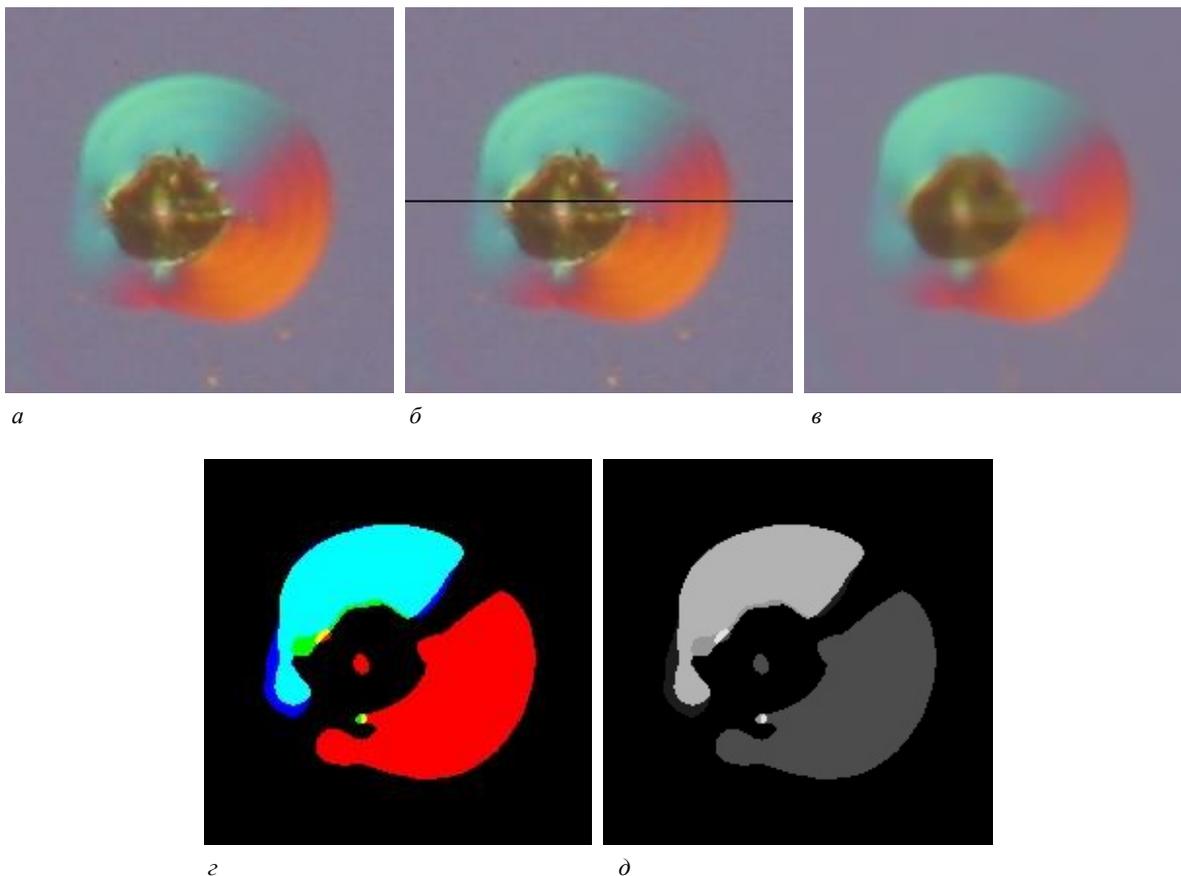
расчет прочностных характеристик материала. В качестве среды разработки программного продукта был выбран язык программирования Python. Схема обработки и анализа данных цифровых изображений включала в себя следующие этапы:

- анализ входных данных изображения;
- обработка входных данных;
- использование медианного фильтра;
- применение функции пороговых значений;
- поиск контуров изображения;
- отображение контуров на изображении;
- вычисление геометрических параметров объекта фотографирования.

Рассмотрим более подробно этапы обработки изображения.

Входные данные изображения. Входными данными являются фотоизображения отпечатков при индентировании полимера, нанесенного на кремниевую подложку (рисунок 1, *a*). Изображение получено методом дифференциально-интерференционного контраста [2], позволяющего выявить неровности поверхности полимера. В центре наблюдается отпечаток пирамиды индентора в полимерной пленке, окруженный навалами полимера. Зона навалов имеет сферическую форму. Изображение состоит из набора пикселей. Если представить изображение в виде сетки, то каждый квадрат в сетке содержит один пиксель, где квадрат с координатами $[0, 0]$ – это левый верхний пиксель. Размер изображения составляет 200×200 пикселей, что в совокупности составляет 40000 пикселей. Все пиксели в изображении представлены в цветовом пространстве RGB (red, green, blue – красный, зеленый, синий), где одно значение для красной компоненты, одно для зеленой и одно для синей. Каждая из трех компонент представлена целым числом в диапазоне от 0 до 255 включительно, которое указывает как «много» цвета содержится.

Если провести линию по центру изображения (рисунок 1, *b*) и взять все значения компонент RGB, можно получить гистограмму значений цвета (рисунок 2, *a*). На гистограмме можно увидеть, как изменяются значения компонент в зависимости от цветов на этой линии.

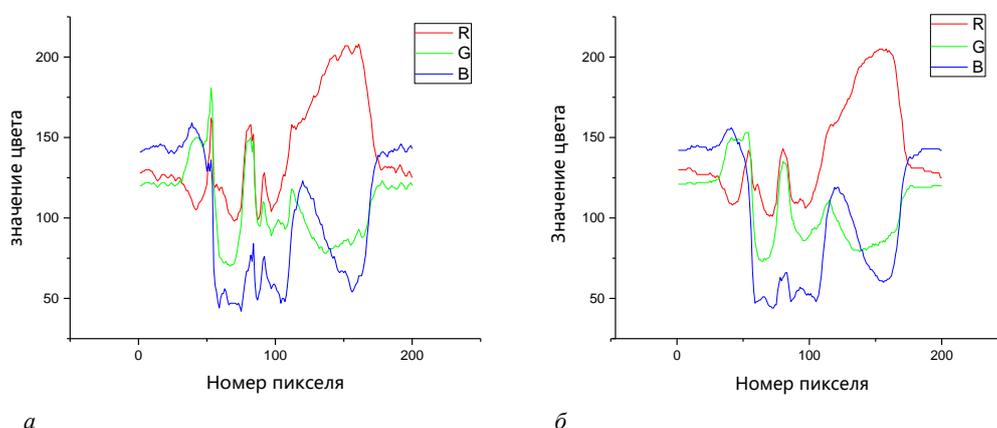


a – входное изображение; *b* – изображение после использования медианного фильтра;
в – изображение с линией, по которой берется значение компонент;
г – изображение после применения функции пороговых значений; *д* – бинарное изображение

Рисунок 1. – Этапы обработки входного изображения

Обработка входных данных. Для обработки данных изображений выбрана библиотека OpenCV и язык программирования Python [3]. OpenCV (Open Source Computer Vision Library) – одна из самых популярных библиотек для приложений по компьютерному зрению. OpenCV-Python – это Python-версия интерфейса для OpenCV. Наличие кода на языке C/C++ данной версии библиотеки в бэкенде гарантирует быстроту ее работы, а Python-обертка во фронтенде обеспечивает легкость настройки и развертывания. Благодаря этому OpenCV-Python является отличным решением для высоконагруженных вычислительных программ по компьютерному зрению. Также в библиотеке содержится огромное количество алгоритмов, есть интерфейсы для многих языков программирования – Java, Ruby, Matlab, Lua, C/C++.

Фильтрация исходных данных. Входное изображение содержит шум, который мешает анализу изображения. Существует много различных типов шумов, таких как гауссовский шум, шум «соли» и «перца» и т.д. Мы можем удалить этот шум из изображения, применив фильтр, который его удаляет или, по крайней мере, сводит к минимуму его влияние. Существует множество различных фильтров, например, усредняющий фильтр (box filter), медианный фильтр (median filter), фильтр типов волн (модовый фильтр, mode filter), фильтр Гаусса (Gaussian filter) и др. Для удаления шума в нашем алгоритме обработки используем медианный фильтр. Медианный фильтр – один из видов цифровых фильтров, который широко используется в цифровой обработке сигналов и изображений для уменьшения уровня шума. Медианное сглаживание применяется в алгоритмах обнаружения краев, потому что при определенных условиях оно сохраняет края, удаляя шум. В результате применения фильтра гистограмма значений RGB стала более ровной (рисунок 2, б).



a – цвета на исходном изображении; *б* – цвета на изображении после фильтра

Рисунок 2. – Визуальное представление цветов на изображении

Применение пороговых значений. Для определения контура следа предлагается функция пороговых значений, разделяющая фон и искомый след. Фон характеризуется постоянными значениями цветов, а для следа характерно резкое изменение значений цвета относительно фона. По гистограмме можно определить, где находится фон и нужный след. Для отделения следа от фона используется функция пороговых значений `cv2.threshold()`. Функция принимает изображение, пороговое значение от 0 до 255, максимальное значение, которое присваивается значениям пикселей, тип порогового преобразования. Как указано выше, изображение состоит из компонент RGB и каждая компонента имеет значения от 0 до 255. Если установить в программе пороговое значение 140, то все значения компонент выше 140 примут значения 255, а компоненты со значениями ниже 140 примут значение 0. После применения функции пороговых значений на изображении след имеет более отчетливый контур (см. рисунок 1, з). Т.е. фон стал черным и принял значения 0, а область внутри следа стала более яркой и приняла соответствующее значение.

Поиск контуров. Контур объекта – это его видимый край, который отделяет след от фона. В действительности большинство методов анализа изображений работают именно с контурами, а не с пикселями как таковыми. Библиотека OpenCV реализует удобные методы для детектирования и манипуляции с контурами изображения. Для поиска контуров используется функция `cv2.FindContours()`. Данная функция принимает три значения: изображение, режим группировки и метод упаковки:

1) *режимы группировки:*

CV2.RETR.LIST – выдает все контуры без группировки;

CV2.RETR.EXTERNAL – выдает только крайние внешние контуры;

CV2.RETR.CCOMP – группирует контуры в двухуровневую иерархию. На верхнем уровне – внешние контуры объекта. На втором уровне – контуры отверстий, если таковые имеются. Все остальные контуры попадают на верхний уровень;

CV2.RETR.TREE – группирует контуры в многоуровневую иерархию;

2) *методы упаковки:*

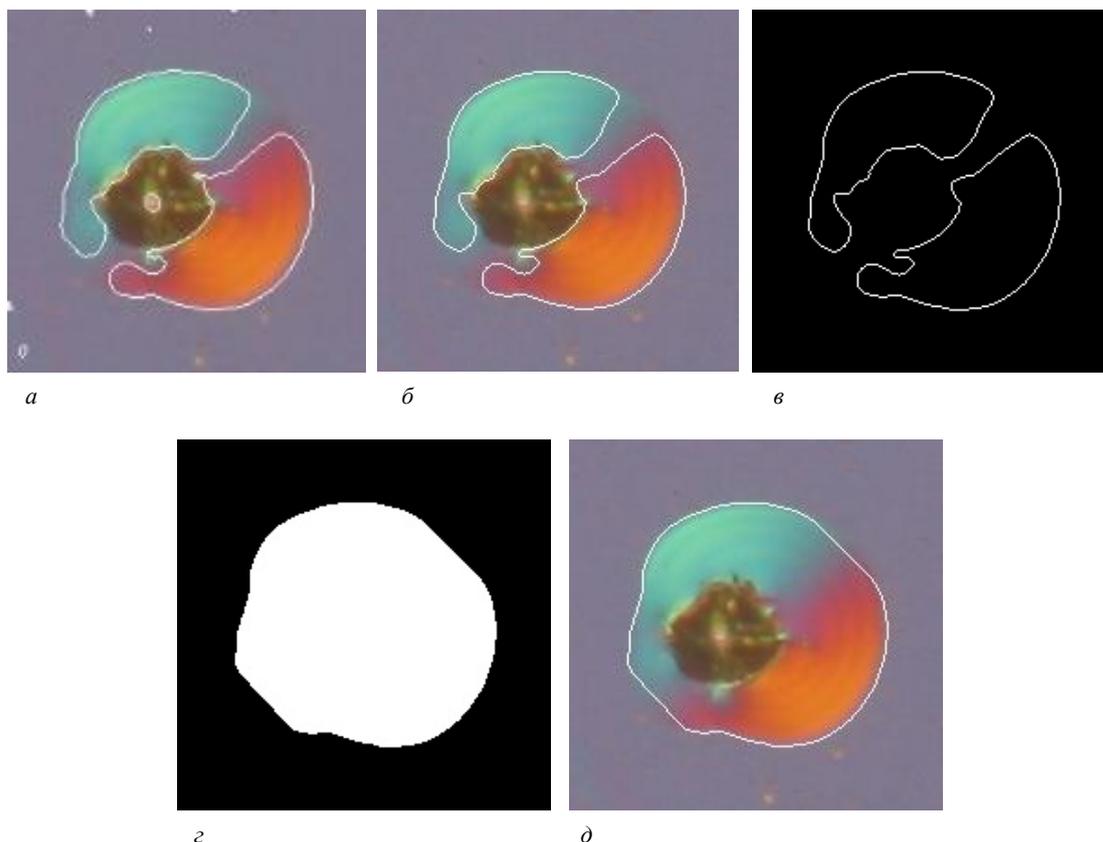
CV.CHAIN.APPROX.NONE – упаковка отсутствует и все контуры хранятся в виде отрезков, состоящих из двух пикселей;

CV2.CHAIN.APPROX.SIMPLE – склеивает все горизонтальные, вертикальные и диагональные контуры;

CV2.CHAIN_APPROX_TC89_L1, CV2.CHAIN_APPROX_TC89_KCOS – применяет к контурам метод упаковки (аппроксимации) Teh-Chin.

Согласно своему формату функция поиска контуров *cv2.FindContours()* принимает изображение, представленное в цветовом пространстве градаций серого (полутонное). Чтобы перевести изображение в цветовое пространство градаций серого применяется функция *cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)*. Данная функция с использованием специальных флагов преобразовывает изображение в другое цветовое пространство согласно заданному флагу. В библиотеке OpenCV доступно более 150 флагов преобразования. После применения функции поиска контуров изображение готово для обработки (см. рисунок 1, д). Полученные контуры следа нужно отобразить на исходном изображении при помощи функции *cv2.drawContours()*. Функция принимает изображение, контур, цвет контура (заданный в палитре RGB), толщину контура в пикселях.

Результат отображений контура на изображении представлен на рисунке 3, а. На изображении выделилось несколько небольших ненужных контуров, которые следует убрать. При отображении контуров на изображении их необходимо отсортировать по площади. При помощи функции *cv2.contourArea()* можно получить значение площади. Функция принимает все контуры и выдает значение площади в пикселях, после чего нужно убрать все ненужные контуры, отсортировав их по площади. Функция площади задается выражением $S = cv2.contourArea(contours[i])$.



а – изображение с нанесенным контуром;
б – изображение с нанесенным контуром после сортировки контуров по размеру;
в – след полученного контура на черном фоне;
г – след объекта на черном фоне после соединения контуров;
д – исходное изображение с нанесенным контуром

Рисунок 3. – Этапы поиска и отображения контуров на исходном изображении

Из-за неравномерного освещения мы получили несколько контуров (см. рисунок 3, в), которые нужно объединить в один. Контур состоит из точек, и при соединении точек контура получили объект, который совпал со следом (см. рисунок 3, г). Полученный след совпадает с исходным обрабатываемым

изображением (см. рисунок 1, а), контур к которому является целью обработки. Для получения полного контура остается применить функцию поиска контуров и отобразить полученный контур на исходном изображении. Результат показан на рисунке 3, д.

Реализация данного алгоритма поиска контура может быть использована для определения геометрических параметров отпечатков при индентировании полимеров, что дает возможность рассчитать некоторые прочностные характеристики:

– оконтуривание отпечатка дает возможность определить две диагонали отпечатка, набрать статистику распределения отпечатков, определить среднее значение и рассчитать микротвердость полимера [4];

– определение контура радиальной трещины позволяет найти длину трещины, статистическое распределение длин трещин и с учетом величины микротвердости рассчитать трещиностойкость K_{1c} (коэффициента интенсивности напряжений) материала [5];

– нахождение размеров разрушенной области позволяет рассчитать удельную энергию отслаивания пленки, являющуюся характеристикой адгезии полимерной пленки к основе [6].

Заключение. Таким образом, реализован алгоритм обработки цветных изображений, полученных при фотографировании поверхности полимера после испытаний на микротвердость, имеющий своей целью определение геометрических размеров отпечатков, контуров областей навалов, длин трещин и других объектов. Это имеет существенное значение для объективного определения прочностных характеристик материалов и для автоматизации процесса измерения в материаловедении.

ЛИТЕРАТУРА

1. Литвинов, Ю.М. Методология определения механических свойств полупроводниковых материалов с помощью метода непрерывного вдавливания индентора / Ю.М. Литвинов, М.Ю.Литвинов // Известия вузов. Материалы электронной техники. – 2004. – № 4. – С. 11–16.
2. Анисович, А.Г. Оптические эффекты при микроскопии неметаллических материалов / А.Г.Анисович // Литье и металлургия. – 2017. – № 1. – С.110–114.
3. Bradski, G. Learning OpenCV. Computer vision with the OpenCV library / G.Bradski, A.Kaehler // O'Reilly Media, Inc. – 2008. – 580 p.
4. Физико-механические свойства облученных пленок диазохинон-новолачного фоторезиста на кремнии / С.А.Вабищевич [и др.] // Вестн. Полоц. гос. ун-та. Сер. С, Фундам. науки. – 2020. – № 12. – С. 60–64.
5. Прочностные свойства структур фоторезист – кремний, γ -облученных и имплантированных ионами B^+ и P^+ / С.А. Вабищевич [и др.] // Вестн. Полоц. гос. ун-та. Сер. С, Фундам. науки. – 2016. – № 12. – С. 30–36.
6. Adhesion of diazoquinon-novolac photoresist films with implanted boron and phosphorus ions to single-crystal silicon / S.A. Vabishchevich [et al.] // High energy chemistry. – 2020. – V. 54, № 1. – P. 46–50.

Поступила 12.05.2021

IMAGE PROCESSING FOR DETERMINING STRENGTH PARAMETERS OF POLYMER FILMS

S. RAHOUSKI, V. TANANA, N. VABISHCHEVICH, S. VABISHCHEVICH

The issues of using digital processing of images of the polymer surface during microhardness tests to determine the geometric dimensions of indentations, zones of destruction and deformation are considered. An image processing algorithm has been built and an image processing program has been implemented. The simulation results can be used when testing polymer films for microhardness to determine the strength characteristics: microhardness, crack resistance, specific peeling energy of the film.

Keywords: *image processing, programming, surface, microindentation, strength characteristics.*