

УДК 004.932

АЛГОРИТМ ДЕТЕКТИРОВАНИЯ УСТАЛОСТНОЙ ТРЕЩИНЫ И ОПРЕДЕЛЕНИЯ МЕСТОПОЛОЖЕНИЯ ЕЁ ВЕРШИНЫ НА ОПТИЧЕСКИХ ИЗОБРАЖЕНИЯХ

С. В. Панин^{1,2}, В. О. Чемезов¹, П. С. Любутин^{1,2}, В. В. Титков¹

¹*Институт физики прочности и материаловедения СО РАН,*

634055, г. Томск, просп. Академический, 2/4

²*Томский политехнический университет,*

634050, г. Томск, просп. Ленина, 30

E-mail: svp@ispms.ru

Предложен и протестирован алгоритм детектирования трещины на оптических изображениях, зарегистрированных при усталостных испытаниях материалов. Алгоритм предназначен для использования при автоматизации измерения параметров роста трещины и слежения за её вершиной в процессе циклического нагружения в целях перемещения оптической системы относительно поверхности исследуемого образца в «область интереса». Установлено, что координаты фрагмента изображения, содержащего трещину, могут быть определены со средней ошибкой 1,93 % от полного размера растра. Тестирование алгоритма на модельных изображениях показало, что средняя ошибка определения местоположения вершины трещины не превышает 56 пикселей.

Ключевые слова: обработка изображений, оптический поток, усталостная трещина, контурный анализ, математическая морфология.

DOI: 10.15372/AUT20170306

Введение. В рамках механики усталостного разрушения одной из ключевых является задача оценки параметров распространения трещины (длины, скорости роста, раскрытия берегов и т. д.), поэтому актуальна проблема автоматизации процесса их измерения, что обуславливает необходимость слежения за её вершиной. Представляют интерес вопросы усталостного разрушения и взаимного влияния накопления повреждённости и эволюции напряжённо-деформированного состояния в окрестности вершины трещины [1]. В частности, подход к описанию основных закономерностей деформирования и накопления повреждений, а также математические модели указанных процессов приведены в [2].

В настоящее время широкое распространение получили оптические системы, позволяющие получать картины распределения деформации в области вершин усталостных трещин. К их числу относятся коммерческие системы VIC 3D, Strain Master [3, 4] и др., преимущественно применяемые в режиме макросъёмки. С позиции оценки сопротивления материалов зарождению и росту усталостных трещин [5] значительный интерес представляют условия, когда длина трещины не превышает десятков микрон. В этом случае применение макрорежима видеофиксации не позволяет достичь требуемого пространственного разрешения [6].

Для решения данной проблемы используются оптические микроскопы, оснащённые фото- или видеокамерой. Однако при больших увеличениях оптической системы в процессе роста усталостной трещины её вершина будет постоянно выходить из поля зрения, поэтому необходимо создавать аппаратно-программные комплексы, позволяющие (в режиме реального времени) управлять механическим сканером, перемещающим микроскоп и видеокамеру относительно образца, на поверхности которого контролируется рост трещины.

Ключевыми работами, направленными на автоматическое детектирование трещин на оптических изображениях, можно считать [7, 8]. В последние годы сформировалось на-

правление прикладных исследований, нацеленных на автоматическое выделение трещин на изображениях. Значительный практический интерес представляет детектирование и количественная характеристика трещин на поверхности дорожного полотна [8–10], заметная часть публикаций посвящена оценке состояния инженерных конструкций и выявлению трещин в них [11–13].

Задачей, решаемой в данной работе, является создание алгоритма детектирования трещины на оптических изображениях поверхности нагруженного материала и определение координат её вершины. Для этого требуется найти координаты фрагмента изображения, содержащего вершину трещины, размер которого не должен превышать 20 % от ширины исходного растра, что позволит автоматизировать процесс позиционирования оптической системы при слежении за трещиной. Кроме того, необходимо обеспечить величину ошибки определения координат вершины трещины, не превышающую 15 % от истинного значения. Это открывает перспективу использования метода корреляции цифровых изображений для получения количественной оценки деформации в процессе циклических испытаний как по берегам усталостной трещины, так и в области её вершины. В результате при больших увеличениях оптической системы можно будет рассчитывать величину раскрытия трещины на разных удалениях от её вершины [14]. Также планируется при испытаниях с переменной амплитудой цикла (либо при случайном спектре нагружения) следить за ростом трещины в автоматическом режиме [15].

1. Выделение трещины. Алгоритм детектирования трещины и определения координат её вершины на серии изображений представлен на рис. 1. Он содержит следующие основные этапы: 1) анализ последовательности изображений (блоки 1–4); 2) выделение



Рис. 1. Алгоритм выделения трещины и определения координат её вершины

объекта максимальной площади, соответствующего области наибольших неравномерностей на картине оптического потока (блоки 5–7); 3) анализ индивидуальных изображений (блоки 8–11); 4) построение скелетона трещины (блок 12); 5) поиск вершины трещины (блок 13).

1.1. *Анализ последовательности изображений.* При анализе серии изображений (либо видеопоследовательности), отображающих распространение трещины, ключевую информацию содержат данные о перемещениях участков (фрагментов). В этом случае характер текстуры на поверхности важен лишь для повышения достоверности (помехоустойчивости и точности) определения перемещений. Для последующего анализа используются данные об оптическом потоке $F(I_t, I_{t-1})$ (I_t и I_{t-1} — изображения в текущий и предыдущий моменты времени), обусловленном ростом несплошности. Таким образом, объектом анализа является оптический поток, визуализация которого в виде цветового кодирования показана на рис. 2, а [16]. При этом разрыв или резкое изменение величин и направлений перемещений в его пределах (на берегах раскрывающейся трещины) свидетельствует о наличии трещины.

Оптический поток можно определить с помощью практически любого из разработанных в последние годы алгоритмов, которые представлены, например, в открытой базе MiddleBury [16]. Однако с точки зрения детектирования трещины алгоритм также должен обладать устойчивостью при поиске перемещений фрагментов изображения на её границах (берегах). Данное требование, предъявляемое к подобного рода алгоритмам, обосновано необходимостью точнее выделять область интереса (здесь и далее под этим термином подразумевается фрагмент изображения, содержащий трещину). В таком случае могут быть использованы алгоритмы, основанные на расчёте взвешенной меры подобия блоков изображений, поскольку они позволяют в полном объёме «сохранять» границы между объектами.

Поскольку у «коротких» трещин (длиной до десятков микрон) величина раскрытия может составлять единицы пикселей, с целью дополнительного контрастирования (подчёркивания границ) применяется процедура дифференцирования поля оптического потока ($dF(I_t, I_{t-1})$) (рис. 2, б), что реализуется с помощью оператора Собеля для окна размером 3×3 [17].

1.2. *Выделение объекта максимальной площади.* Для определения области интереса полученное изображение дифференцированного оптического потока подвергается нормализации (приведению к диапазону $[0, 255]$) и пороговой бинаризации. В качестве порога бинаризации принимается значение яркости равное единице, т. е. «отбрасываются» только самые малые перемещения. Далее к бинаризованному изображению применяется операция морфологического замыкания [17] в целях заполнения мелких промежутков внутри сплошных объектов. Процесс завершается поиском контуров оставшихся на изображении объектов и оценкой их формы. В качестве трещины принимается объект наибольшей площади. Данный метод используется для первичного (грубого) детектирования фрагмента изображения, наиболее вероятно содержащего трещину (рис. 2, в).



Рис. 2. Результаты анализа последовательности изображений на основе определения поля перемещений: а — изображение оптического потока; б — изображение дифференцированного оптического потока; в — алгоритмически определённая локализованная трещина

1.3. *Анализ индивидуальных изображений.* При обработке одиночных изображений (без учёта остальных изображений в серии/видеопотоке) (рис. 3, *a*) в основе анализа данных лежит информация о текстуре поверхности. В процессе слежения за трещиной решаются две ключевые задачи: её поиск на изображении (детектирование) и последующее определение местоположения её вершины (локация). Для этого проводится контурный анализ, основанный на разбиении изображения на отдельные объекты и фон (рис. 3, *b*), т. е. процедура сегментации [17]. Такой анализ позволяет выделить объекты, максимально подобные трещине (рис. 3, *c*).

Априорно считается, что трещина (как область, соответствующая отсутствующему материалу — несплошности) должна являться наиболее тёмным участком (объектом) на изображении. В целях снижения вычислительных затрат и повышения устойчивости работы алгоритма детектирования трещин используется процедура бинаризации, позволяющая заметно сократить количество выделяемых и рассматриваемых объектов. Для бинаризации применяется алгоритм, в котором для вычисления глобального порогового значения яркости анализируется соответствующая гистограмма. Применение глобального порога бинаризации обусловлено тем, что локально определяемый порог не даёт возможности с

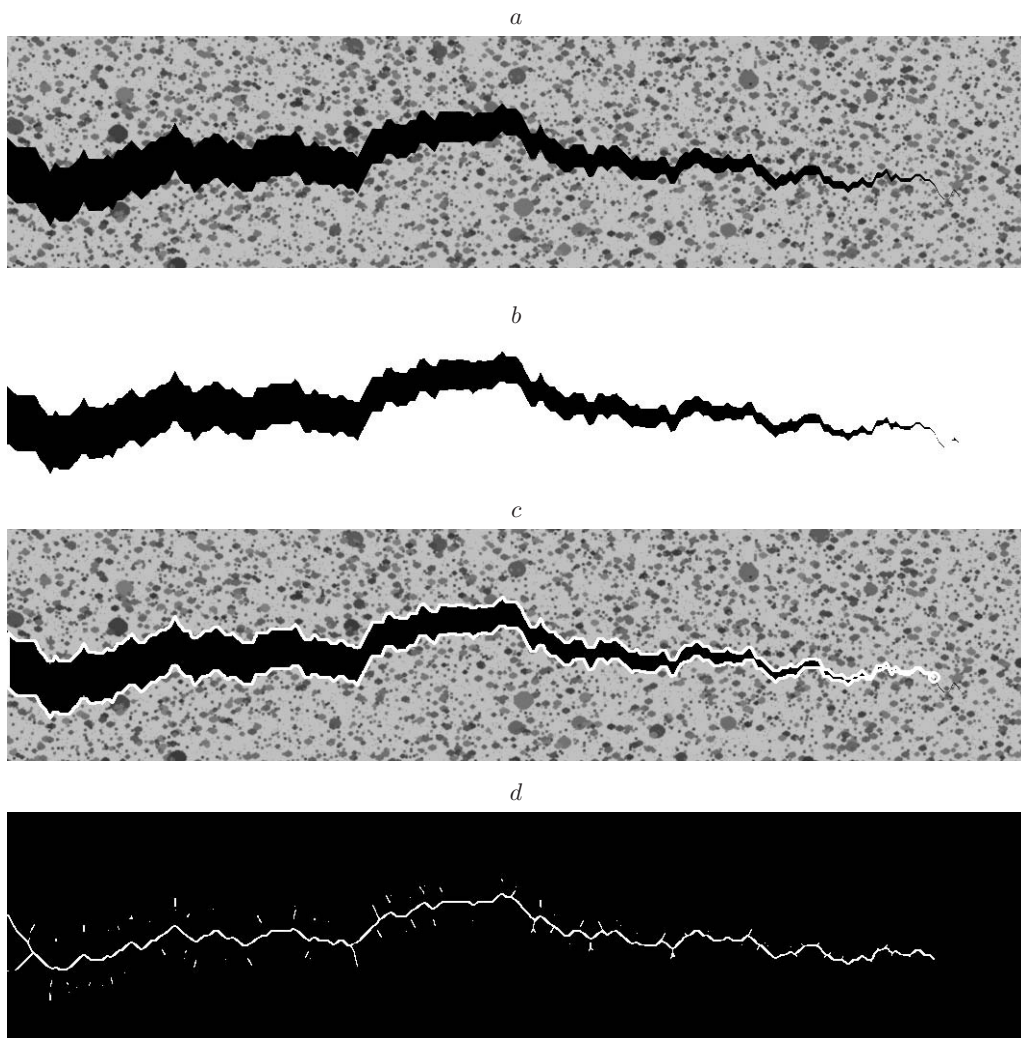


Рис. 3. Детектирование трещины при анализе индивидуального изображения на основе выделения контуров: *a* — входное изображение; *b* — бинаризованное изображение для сегментации; *c* — выделенная трещина; *d* — найденный скелетон трещины

приемлемой чувствительностью выделить области с однородной яркостью. В качестве глобального порога принимается максимум гистограммы в области малых яркостей (область гистограммы с уровнем яркости ниже 64; величина данного параметра была подобрана эмпирическим путём).

Последующей операцией в обработке полученного чёрно-белого изображения является морфологическое замыкание [16]. Его целью является удаление мелких (тёмных) не относящихся к трещине фрагментов на изображении (формо- и размерозависимая фильтрация), соответствующих элементам текстуры на поверхности. Затем полученное чёрно-белое изображение подвергается контурному анализу для детектирования объекта (объектов), геометрические характеристики которого максимально соответствуют таковым для трещины. В качестве трещины принимается объект максимальной площади.

1.4. *Построение скелетона трещины.* Для поиска и определения координат вершины несплошности используется построение скелетона [18] объекта типа «трещина». Для этого оценивается кратчайшее расстояние от точки внутри трещины до её границ (рис. 3, *d*). Эффективность применения данной операции связана с тем, что скелетоном несплошности является кривая, объединяющая точки, равноудалённые от её границ.

1.5. *Поиск вершины трещины.* Для определения координат вершины несплошности проводится оценка величины раскрытия вдоль оси скелетона. Решение данной задачи упрощается тем, что все смещения уже были рассчитаны в первой части работы алгоритма. Искомая точка принадлежит участку, характеризующему наименьшими смещениями.

2. Тестирование алгоритма. 2.1. *Получение модельных изображений.* Тестирование алгоритма проводили с использованием серии модельных изображений. Функционирование предложенного алгоритма на первом этапе исследовали в условиях отсутствия возмущающих воздействий (шум, размытие, имитирующее расфокусировку, и др.). Серию модельных изображений формировали в несколько этапов в виде текстурированной поверхности, имитирующей напылённый спекл [14, 19], на которую программным способом накладывалась трещина (рис. 4, *a, b*): 1) создание модели поверхности материала

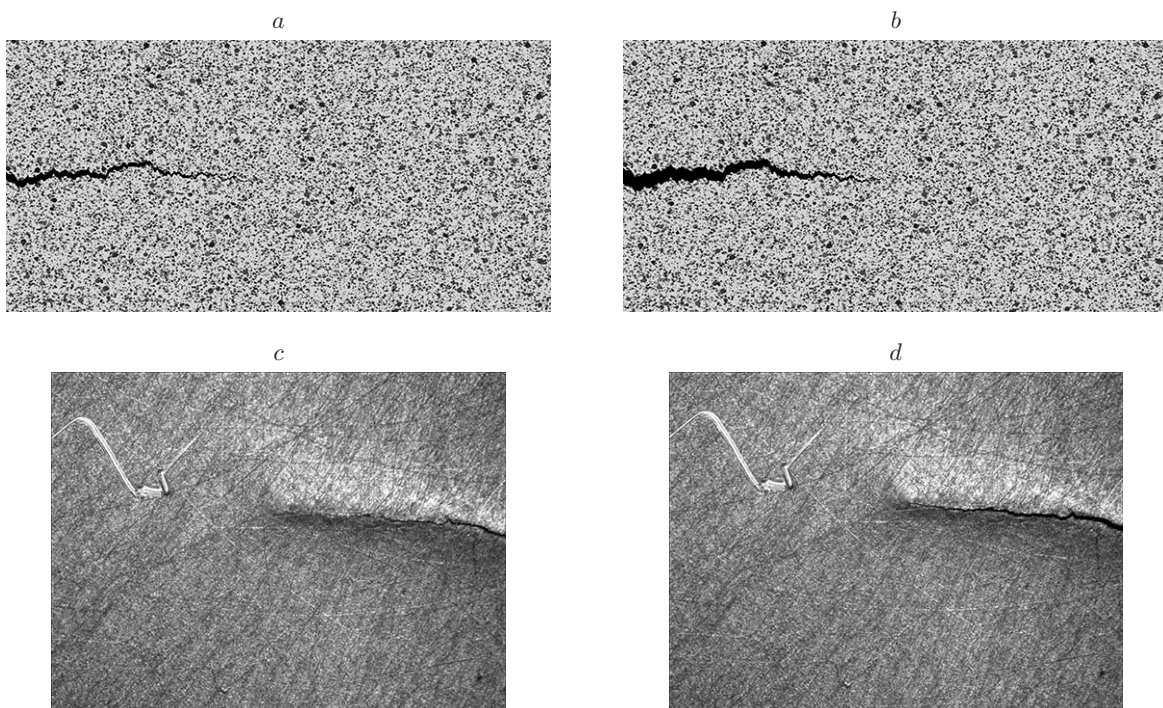


Рис. 4. Изображения, характеризующие распространение усталостной трещины: *a, b* — модельные, *c, d* — полученные экспериментально

(формирование изображения спекла), 2) задание параметров несплошности (длина, раскрытие, форма берегов, имитирующая её ветвление), 3) изменение модели поверхности (пересчёт изображения) с учётом изменения параметров трещины.

Для определения устойчивости работы алгоритма на втором этапе тестирования было сгенерировано восемь пар модельных изображений, отражающих четыре варианта траектории роста трещины и два типа текстуры. Каждая траектория получена случайным разбиением линии раскрытия трещины. Длина последней составляла половину ширины изображения, а величина раскрытия между двумя соседними изображениями в последовательности, соответствующими моментам времени t и $t + 1$, составляла 30 и 60 пикселей. Используемые варианты текстуры различались размером напылённого спекла, когда максимальный диаметр пятен равнялся 20 и 30 пикселей. Сгенерированные модельные изображения подвергали размытию и зашумлению (для имитации искажений, характерных для лабораторных экспериментов).

Зашумление изображений проводили путём наложения на каждый пиксел (значения яркости которого лежат в диапазоне $[0; 255]$) случайного значения, имеющего нормальное распределение (что подобно характеру шумов для реальных видеодатчиков) и различное стандартное отклонение $s = [0; 45]$. Для размытия применяли нормальное гауссово размытие со среднеквадратичным отклонением $\sigma = [0; 4,5]$.

Дополнительно проведены эксперименты по детектированию трещины при обработке экспериментальных изображений [6], зарегистрированных при циклическом растяжении образца алюминиевого сплава Д16 (рис. 4, *c, d*). Для оценки эффективности предложенного в данной работе алгоритма были использованы два критерия:

1. Ошибка определения местоположения вершины трещины, равная расстоянию между её координатами на модельном изображении (с их известным значением) и координатами, найденными с помощью предложенного алгоритма:

$$D = \sqrt{(C_{xr} - C_{xf})^2 + (C_{yr} - C_{yf})^2},$$

где (C_{xr}, C_y) и (C_{xf}, C_{yf}) — модельные и найденные координаты соответственно. Заметим, что причиной данной ошибки в первую очередь может являться потеря информации в области вершины трещины (см. рис. 3, *b*), когда алгоритм в процессе бинаризации «отсекает» фрагмент, также относящийся к несплошности.

2. Ошибка определения площади трещины, выделенной с помощью предложенного алгоритма (см. разд. 1.2), которая рассчитывается из отношения найденной площади S_f к её величине S_r , заданной в модели:

$$dES = |S_f/S_r - 1| \cdot 100 \text{ \%}.$$

О повышении точности функционирования алгоритма должно свидетельствовать достижение минимального значения каждого из этих двух показателей.

2.2. *Результаты тестирования и их обсуждение.* На рис. 5 представлены графики зависимости ошибок расчёта параметров трещины от степени зашумления s (рис. 5, *a*) и размытия σ (рис. 5, *b*) исходных изображений. Показано, что для незашумлённых изображений (при стандартном отклонении шума и среднеквадратичном отклонении фильтра размытия равных нулю) средняя ошибка определения площади трещины составила 1,93 %, а координат её вершины — 56 пикселей. При этом стандартное отклонение определения площади трещины и координат её вершины составляет 0,21 % и 15 пикселей соответственно (согласно приведённым на графиках величинам разброса анализируемых параметров).

2.3. *Зашумление изображений.* Результаты анализа всех восьми пар изображений представлены графиками на рис. 5, *a* и показывают, что использование предложенного

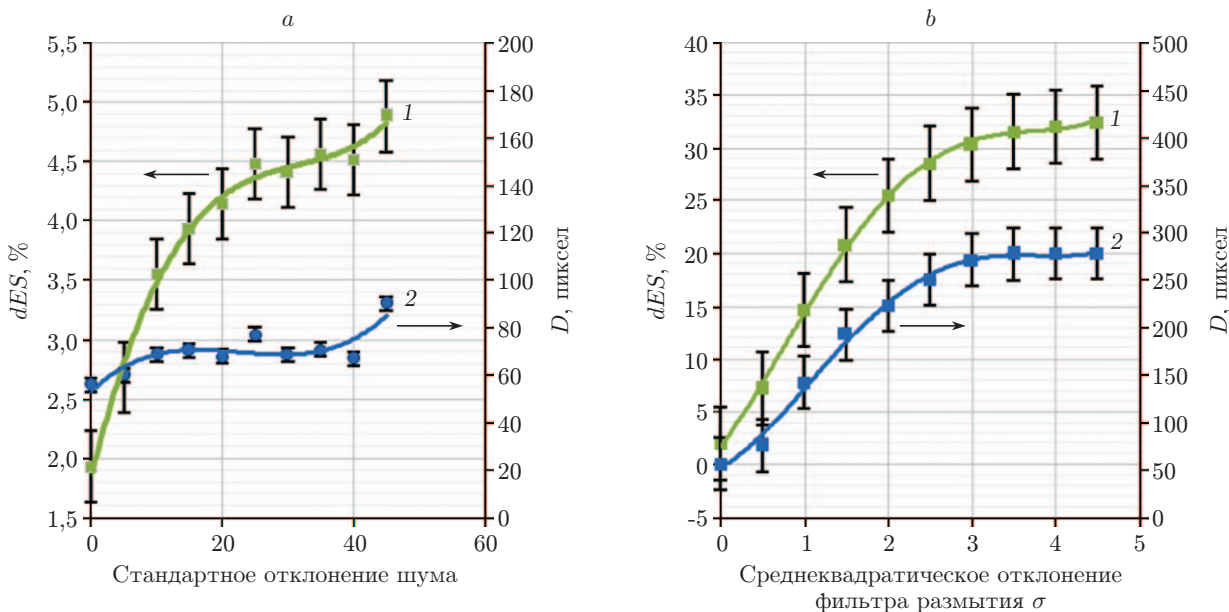


Рис. 5. Графики зависимости ошибок определения площади трещины и координат её вершины: *a* — от уровня шума, *b* — от размытия. Сплошные кривые — аппроксимация, 1 — ошибка определения площади трещины dES , 2 — ошибка определения местоположения вершины трещины D

алгоритма в условиях значительного зашумления ($s = 45$) обуславливает максимальную величину ошибки при детектировании трещины ($dES = 5,2$ %) (кривая 1). При определении координат её вершины (кривая 2) возрастание ошибки происходит практически линейно и максимум данного параметра $D = 90,2$ (с учётом разброса) также наблюдается при максимальном стандартном отклонении шума ($s = 45$). По мнению авторов, в рамках задачи слежения за трещиной при позиционировании оптической системы относительно поверхности образца с учётом поставленных в начале данной работы технических требований подобные величины являются вполне приемлемыми.

2.4. *Размытие изображений* более критично влияет на величину ошибки (см. рис. 5, *b*), которая возрастает в значительно большей степени и достигает $dES = 36$ % при размытии фильтром с $\sigma = 4,5$ (кривая 1). При определении координат вершины трещины (кривая 2) возрастание ошибки происходит подобным образом; максимум данного параметра $D = 280,2$ также наблюдается при максимальном среднеквадратичном отклонении фильтра размытия $\sigma = 4,5$.

В постановке задачи исследований было указано, что точность определения участка, содержащего вершину трещины, должна составлять порядка 20 % от горизонтального размера раstra, т. е. при изображениях размером не менее 2000×1000 пикселей максимально допустимая величина ошибки определения координат составляет 400 пикселей. В случае всего исследованного диапазона зашумления (см. рис. 5, *a*) порог не оказывается превышенным, в то время как для размытия уже при $\sigma \geq 3$ ошибка алгоритма превышает допустимое значение (см. рис. 5, *b*). Это обусловлено тем, что значительная часть трещины имеет расстояние между берегами большее, чем апертура фильтра размытия. Данное обстоятельство, в свою очередь, накладывает ограничения на минимальное значение раскрытия трещины, которое может быть найдено с помощью предложенного алгоритма.

С одной стороны, в предлагаемой работе введено ограничение на максимальную ошибку определения координат вершины трещины, не превышающую 15 % от их «истинного» положения. Результаты детектирования трещины в условиях зашумления полностью удо-

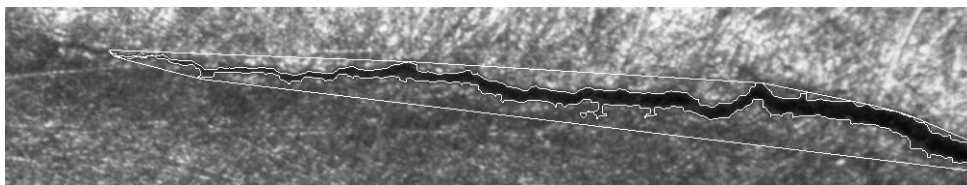


Рис. 6. Контуры трещины, выделенные на экспериментальном изображении с помощью разработанного алгоритма

влетворяют данному условию, так как ошибка не превышает $dES \leq 5,5$ %. С другой стороны, при имитации расфокусировки предельное значение такой ошибки достигается уже при $\sigma > 1$. Это означает, что алгоритм может быть эффективно использован только для случаев небольших расфокусировок оптической системы.

Результат детектирования трещины на экспериментальных изображениях представлен на рис. 6. Видно, что алгоритмически выделенный контур чётко повторяет границы трещины, что свидетельствует в пользу возможности применения алгоритма для решения практических задач выделения трещин на реальных оптических изображениях.

Заключение. В данной работе предложен и протестирован алгоритм автоматического детектирования трещины и определения координат её вершины на оптических изображениях. Тестирование с использованием модельных изображений (включая размытые и зашумлённые) продемонстрировало, что алгоритм позволяет найти координаты вершины трещины со средней ошибкой порядка 56 пикселей. Средняя ошибка определения площади трещины (включая выделение её берегов) не превышает 1,93 %. Результаты тестирования алгоритма на экспериментальных изображениях показали возможности его эффективного применения для решения задач выделения трещин на реальных оптических изображениях.

Предложенный алгоритм характеризуется максимальной ошибкой детектирования трещины в условиях сильного зашумления ($s = 45$) равной 5,7 %. В то же время величина ошибки резко возрастает в условиях размытия фильтром с $\sigma = 4,5$, достигая 40 %. Таким образом, для решения задачи позиционирования оптической системы относительно поверхности циклически нагружаемого образца зашумление изображений в исследованном диапазоне значений не оказывает критического влияния на эффективность работы алгоритма, в то время как расфокусировка заметно лимитирует возможность детектирования трещины и определения координат её вершины.

СПИСОК ЛИТЕРАТУРЫ

1. **Степанова Л. В.** Уточненный расчет напряженно-деформированного состояния у вершины трещины в условиях циклического нагружения в среде с поврежденностью // Вестн. СамГУ. Естественно-научная сер. 2011. № 2(83). С. 105–115.
2. **Волков И. А., Коротких Ю. Г.** Уравнения состояния вязко упругопластических сред с поврежденностью. М.: Физматлит, 2008. 424 с.
3. **The VIC-3D System** / Correlated Solutions, Inc. 2016. URL: <http://www.correlatedsolutions.com/vic-3d/> (дата обращения: 12.10.2016).
4. **2D-/3D Stereo Digital Image Correlation** / LaVision. 2016. URL: <http://www.lavision.de/en/products/strainmaster/strainmaster-dic.php> (дата обращения: 12.10.2016).
5. **Panin V. E., Panin A. V., Elsukova T. F., Popkova Yu. F.** Fundamental role of crystal structure curvature in plasticity and strength of solids // Phys. Mesomech. 2015. **18**, N 2. P. 89–99.
6. **Sunder R., Biakov A., Eremin A., Panin S.** Synergy of crack closure, near-tip residual stress and crack-tip blunting in crack growth under periodic overloads — A fractographic study // Intern. Journ. Fatigue. 2016. **93**, Pt. 1. P. 18–29.

7. **Song K. Y., Petrou M., Kittler J.** Texture crack detection // *Machine Vision and Appl.* 1995. **8**, Is. 1. P. 63–75.
8. **Kirschke K., Velinsky S.** Histogram-based approach for automated pavement-crack sensing // *Journ. Trans. Eng.* 1992. **118**, Is. 5. P. 700–710.
9. **Zou Q., Cao Y., Li Q. et al.** CrackTree: Automatic crack detection from pavement images // *Pattern Recogn. Lett.* 2012. **33**, Is. 3. P. 227–238.
10. **Sun L., Kamaliardakani M., Zhang Y.** Weighted neighborhood pixels segmentation method for automated detection of cracks on pavement surface images // *Journ. Comput. in Civil Eng.* 2016. **30**, Is. 2. 04015021.
11. **Yamaguchi T., Hashimoto S.** Fast crack detection method for large-size concrete surface images using percolation-based image processing // *Machine Vision and Appl.* 2010. **21**, Is. 5. P. 797–809.
12. **Iyer S., Sinha S. K.** A robust approach for automatic detection and segmentation of cracks in underground pipeline images // *Image and Vision Comput.* 2005. **23**, Is. 10. P. 921–933.
13. **Lins R. G., Givigi S. N.** Automatic crack detection and measurement based on image analysis // *IEEE Trans. Instrum. and Meas.* 2016. **65**, Is. 3. P. 583–590.
14. **Sutton M. A., Orteu J.-J., Schreier H.** *Image Correlation for Shape, Motion and Deformation Measurements: Basic Concepts, Theory and Applications.* Springer, 2009. 321 p.
15. **Abdel-Qader I., Abudayyeh O., Kelly M.** Analysis of edge-detection techniques for crack identification in bridges // *Journ. Comput. in Civil Eng.* 2003. **17**, Is. 4. P. 255–263.
16. **Baker S., Scharstein D., Lewis J. et al.** A database and evaluation methodology for optical flow // *Computer Vision.* 2011. **92**, N 1. P. 1–31.
17. **Gonzalez R. C., Woods R. E.** *Digital Image Processing.* Upper Saddle River, USA: Prentice Hall, 2002. 793 p.
18. **Glantz R., Kropatsch W. G.** Skeletons in the framework of graph pyramids // *Lecture Notes in Comput. Sci.* 2001. **2059**. P. 186–195.
19. **Панин С. В., Титков В. В., Любутин П. С.** Выбор параметров алгоритма трёхмерного рекурсивного поиска при построении поля векторов перемещений с использованием иерархического подхода // *Автометрия.* 2015. **51**, № 2. С. 27–37.

Поступила в редакцию 12 октября 2016 г.