

## АЛГОРИТМ ОБНАРУЖЕНИЯ ИСКАЖЁННЫХ ДУБЛИКАТОВ НА ЦИФРОВЫХ ИЗОБРАЖЕНИЯХ С ИСПОЛЬЗОВАНИЕМ БИНАРНЫХ ГРАДИЕНТНЫХ КОНТУРОВ

А.В. Кузнецов<sup>1,2</sup>, В.В. Мясников<sup>1,2</sup>

<sup>1</sup> Самарский государственный аэрокосмический университет имени академика С.П. Королёва (национальный исследовательский университет) (СГАУ), Самара, Россия,

<sup>2</sup> Институт систем обработки изображений РАН, Самара, Россия

### Аннотация

Встраивание дубликатов является одним из самых очевидных способов намеренного искажения цифровых изображений с целью сокрытия содержащейся в последних информации. Процесс встраивания заключается в копировании фрагмента изображения из одной области в другую область того же изображения. При этом копируемый фрагмент может быть подвержен различным преобразованиям: контрастированию, добавлению шума, масштабированию, повороту, а также различным их комбинациям. Существующие подходы к поиску искажённых областей основаны на вычислении векторов признаков в процессе разбиения изображения на пересекающиеся блоки и использовании этих векторов для поиска близких в евклидовом пространстве областей. В данной работе мы предлагаем использовать бинарные градиентные контура, которые являются устойчивыми к локальному контрастированию, аддитивному шуму и компрессии. В работе также представлены аргументация выбора этого способа для поиска искажённых дубликатов, а также результаты экспериментальных исследований различных вариантов реализации локальных бинарных шаблонов.

**Ключевые слова:** обнаружение дубликатов, искажённый дубликат, локальный бинарный шаблон, бинарные градиентные контура, вектор-признаков, k-d-дерево.

**Цитирование:** Кузнецов, А.В. Алгоритм обнаружения искажённых дубликатов на цифровых изображениях с использованием бинарных градиентных контуров / А.В. Кузнецов, В.В. Мясников // Компьютерная оптика. – 2016. – Т. 40, № 2. – С. 284-293. – DOI: 10.18287/2412-6179-2016-40-2-284-293.

### Введение

В эпоху цифровых технологий невозможно представить себе более популярного средства представления и передачи информации, чем цифровые изображения и видео. Они используются во всех сферах жизнедеятельности и имеют приложение во многих отраслях науки и техники. Так, фотографии применяются повседневно в СМИ, социальных сетях, облачных сервисах хранения данных. Согласно статистике [1], в 2015 году в известной социальной сети Facebook ежедневно загружалось в среднем 300 миллионов цифровых изображений. Следует учесть, что в последние несколько лет появилось огромное количество веб-приложений (Instagram, Twitter, Flickr и т.д.), которые ежедневно осуществляют сопоставимый с Facebook трафик цифровых изображений. Такой колоссальный объём данных не может не интересовать злоумышленников, потому что человеку свойственно сначала воспринимать информацию как она есть, а затем уже подвергать её критике. В зависимости от содержащейся в цифровом изображении информации, она может быть использована в компрометирующей форме и привести к серьёзным политическим и экономическим последствиям, если контрафактность данных не будет обнаружена. Именно поэтому в современном мире цифровые данные должны быть как можно лучше защищены от изменений злоумышленниками, чтобы не быть лёгким оружием в информационном противостоянии.

Одним из самых часто используемых способов подделывания изображения является копирование и вставка области этого же изображения. Такие атаки

называются дублированием фрагментов цифрового изображения, а копируемые области – *дубликатами*. Между копированием и вставкой к дубликату могут применяться различные преобразования: геометрические (масштаб, поворот), яркостные (контрастирование, добавление шума) и другие. Такие дубликаты будем называть *искажёнными*. В рамках данной работы под искажениями будем понимать изменения, возникающие в результате линейного контрастирования, внесения аддитивного шума, компрессии. Очевидно, что внесение изменений в копируемый фрагмент изображения приводит к усложнению обнаружения данного вида атаки.

В настоящее время существует большое количество работ, посвящённых разработке алгоритмов обнаружения искажённых и неискажённых дубликатов: общим в них является разработка некоторых признаков, инвариантных к вносимым в ходе преобразований искажениям, вычисляемых для положений скользящих окон с перекрытиями [2]. Ранее авторами работы были разработаны алгоритмы обнаружения неискажённых дубликатов, в основе которых лежали вычисление значений хэш-функций в скользящем окне и дальнейший поиск одинаковых значений при помощи хэш-таблицы [3–5]. Предложенные решения показали высокую эффективность при низкой вычислительной сложности.

Существующие алгоритмические решения по обнаружению искажённых дубликатов могут быть разбиты на пять основных групп [2]:

1) алгоритмы на основе вычисления моментных признаков (Blur [6], Zernike [7]);

- 2) алгоритмы на основе признаков снижения размерности (PCA [8], SVD [9]);
- 3) алгоритмы на основе анализа локальных яркостных характеристик (Luo [10], Bravo [11]);
- 4) алгоритмы на основе анализа частотных характеристик (DCT [12], FMT [13]);
- 5) алгоритмы, определяющие ключевые точки на изображении (SIFT [14], SURF [15]).

Предлагаемый в данной работе алгоритм относится к группе алгоритмов, использующих для поиска дубликатов признаки, вычисляемые на основе анализа локальных яркостных характеристик (группа 3). Одним из способов вычисления локальных яркостных характеристик является построение локальных бинарных шаблонов (ЛБШ, *Local Binary Pattern*). ЛБШ ранее уже использовались для обнаружения искажённых дубликатов [16–17]. Тем не менее в работе по классификации текстур [18] было отмечено, что ЛБШ являются неустойчивыми к шуму и необходимо использовать более устойчивые вариации локальных шаблонов для построения признаков, инвариантных к разного рода искажениям.

В данной работе мы предлагаем новый алгоритм обнаружения искажённых дубликатов на основе вычисления признаков с использованием бинарных градиентных контуров (БГК), предложенных в работе [19] А. Fernández и коллективом соавторов для решения задачи классификации текстур. БГК являются модификацией ЛБШ, и их использование позволяет с высокой точностью обнаруживать дубликаты с внесёнными искажениями, что подтверждается результатами экспериментальных исследований.

Работа построена следующим образом. В первом пункте дано описание ЛБШ и его распространённых форм, используемых для повышения устойчивости к искажениям: стандартный ЛБШ, устойчивый к повороту ЛБШ, полный ЛБШ и его модификация, устойчивый ЛБШ и локальный тернарный шаблон (ЛТШ). Второй пункт работы посвящён описанию трёх форм БГК, которые будут применяться в работе. Третий пункт посвящён описанию алгоритма обнаружения дубликатов, в рамках которого используются новые признаки на основе БГК. В четвёртом пункте приводятся результаты экспериментальных исследований разработанного алгоритма обнаружения искажённых дубликатов в зависимости от параметров вносимых искажений.

**1. Локальный бинарный шаблон и его разновидности**

Локальный бинарный шаблон

ЛБШ является достаточно известным и часто используемым в распознавании образов локальным признаком цифровых изображений. Этот вид локальных признаков привлекает внимание исследователей благодаря низкой вычислительной сложности, инвариантности к равномерному изменению яркости и способности описывать текстуры [20, 21]. На практике оператор ЛБШ совмещает свойства статистического и структурного текстурного анализа, позволяя строить дескрипторы цифровых изображений.

Идея ЛБШ принадлежит Wang [20], и была позднее развита Ojala и коллективом соавторов в работах [21, 22], где шаблон использовался в задачах классификации. Впоследствии ЛБШ получили широкое применение в обнаружении и распознавании лиц, в обнаружении движущихся объектов и в других областях анализа цифровых сигналов. Под ЛБШ понимают некоторое описание окрестности пикселя в бинарной форме. В базовой версии для построения ЛБШ использовалось окно 3×3 в окрестности центрального пикселя. 8 соседних значений сравнивались с центральным, и в зависимости от результата сравнения центральному пикселю ставилось в соответствие 8-битное число, которое называется *кодом ЛБШ* (рис. 1) (в данном случае значению “0” соответствует ситуация, когда центральное значение превосходит соседнее, “1” – наоборот).

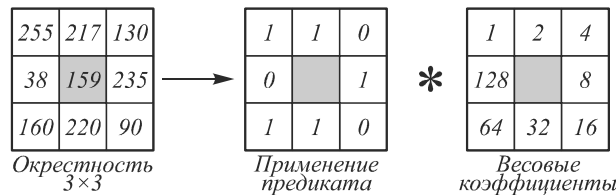


Рис. 1. Базовая версия ЛБШ.  
Код ЛБШ – 1+2+8+32+64=107

По аналогии с этим алгоритмом в дальнейшем стали использоваться более широкие окрестности центрального пикселя для построения кода (рис. 2).

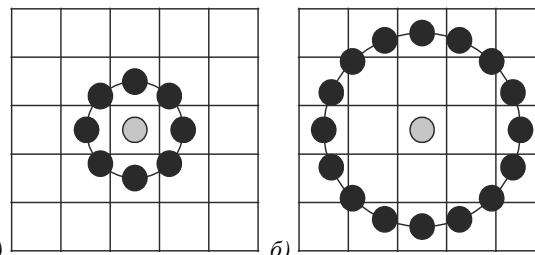


Рис. 2. Центральнo-симметричные окрестности пикселя для различных пар значений (P, R): а) (8, 1) и б) (16, 2)

Определим изображение как функцию яркости пикселя  $f(x, y)$ . Пусть также определён бинарный вектор  $\mathbf{b}$  длины P:

$$\mathbf{b} = (b_0, \dots, b_{P-1}), b_i \in \mathbf{B}, \mathbf{B} = \{0, 1\}.$$

Определим далее для центрального пикселя  $f_c = f(x_c, y_c)$  P равноудалённых соседей  $f_i = f(x_i, y_i)$ ,  $i = 0, P-1$ , расположенных на окружности радиуса R. Тогда код ЛБШ определяется следующим образом:

$$LBP_{P,R}(x_c, y_c) = \sum_{i=0}^{P-1} I(f_i \geq f_c) \cdot 2^i \equiv \sum_{i=0}^{P-1} b_i \cdot 2^i, \quad (1)$$

где I – индикатор события, определяемый следующим образом:

$$I(n) = \begin{cases} 1, & n = true, \\ 0, & n = false. \end{cases}$$

В терминах, введённых в работе В.В. Мясникова [23], выражение (1) называется строгим кодовым преобразованием, так как приводит к взаимно-однозначному

переводу бинарной последовательности в целочисленное значение. Так, например, для 8-компонентного вектора  $\mathbf{b}$  имеем 256 различных значений кодов ЛБШ. Как было описано в работе [18], такое преобразование является чувствительным к шумовым искажениям (квантование вследствие округления при контрастировании, добавление шума и др.), при этом границы чувствительности можно указать явно [23]. Ввиду проблемы чувствительности ЛБШ в прикладных задачах часто используются другие формы ЛБШ, которые будут описаны далее в этом разделе.

Стандартная форма шаблона

Одним из способов решения проблемы чувствительности ЛБШ к шумовым искажениям является использование *стандартных ЛБШ (Uniform LBP)* [24], отличающихся повышенной устойчивостью к указанным выше искажениям. Стандартные ЛБШ позволяют выявлять локальные примитивы, такие как: контура (edge), пятна (spot), углы (corner). Дадим определение кода стандартного ЛБШ. Для этого введём несколько понятий.

Определим функцию количества переходов для бинарного вектора  $\mathbf{b}$ :

$$U(\mathbf{b}) = \sum_{i=0}^{P-2} I(b_i \neq b_{i+1}). \tag{2}$$

Кодом стандартного ЛБШ будем называть код ЛБШ следующего вида:

$$LBP_{P,R}^U(x_c, y_c) = \begin{cases} \sum_{i=0}^{P-1} I(f_i \geq f_c) \cdot 2^i, U(\mathbf{b}) \leq 2, \\ P \cdot (P-1) + 3, \text{ иначе.} \end{cases} \tag{3}$$

Примерами стандартных ЛБШ с параметрами  $(P,R)=(8,1)$  являются битовые вектора 11000111, 11111000, 00000000. Исходя из определения, для  $P$ -битного вектора  $\mathbf{b}$  количество кодов стандартного ЛБШ  $LBP_{P,R}^U$  составляет  $P \cdot (P-1) + 3$ . Достаточно очевидно, что стандартные ЛБШ не являются строгими кодовыми преобразованиями. В частности, если для ЛБШ с параметрами  $(P,R)=(8,1)$  имеем 256 различных кодов, для стандартного ЛБШ имеем только 59 кодов. Такое уменьшение числа кодов позволяет снизить количество ошибок, возникающих вследствие шумовых искажений, но при этом ведёт к росту числа коллизий (ложно обнаруженных дубликатов).

Форма стандартного шаблона, устойчивая к повороту

В работе [22] приводится также устойчивая к повороту форма стандартного ЛБШ, в основе которой лежит выбор минимального значения кода стандартного ЛБШ среди всех его циклических сдвигов. Пусть для некоторого положения  $(x_c, y_c)$  получен код стандартного ЛБШ  $LBP_{P,R}^U(x_c, y_c)$ , тогда код устойчивого к повороту стандартного ЛБШ вычисляется следующим образом:

$$LBP_{P,R}^{RIU}(x_c, y_c) = \begin{cases} \min_k W(LBP_{P,R}^U, k), U(\mathbf{b}) \leq 2 \\ P+1, \text{ иначе} \end{cases}, \tag{4}$$

где  $W(LBP_{P,R}^U, k)$  – функция циклического сдвига кода стандартного ЛБШ на  $k$  бит влево. Преобразование (4) приводит к формированию только  $P+1$  различных значений кодов.

Полная форма шаблона

Отличительной особенностью другой формы ЛБШ, называемой *полной* [25–26], является аккумулярованное описание локальной структуры изображения с использованием трёх составляющих: знаковой, амплитудной и центральной. Эти компоненты могут быть использованы в любой комбинации для построения расширенного кода.

Знаковая составляющая  $CLBP_{S,P,R}$  представляет из себя код ЛБШ вида (1). Амплитудная составляющая представляет из себя ЛБШ, код которого вычисляется следующим образом:

$$CLBP_{-M_{P,R}}(x_c, y_c) = \sum_{i=0}^{P-1} I(|f_i - f_c| \geq E_M) \cdot 2^i, \tag{5}$$

где  $E_M$  – среднее значение амплитуд  $|f_i - f_c|$ . Центральная составляющая позволяет учитывать значение центрального пикселя  $f_c$  при построении кода, и её код, соответственно, выглядит следующим образом:

$$CLBP_{-C_{P,R}}(x_c, y_c) = I(f_c \geq E_f), \tag{6}$$

где  $E_f$  – среднее значение изображения  $f$ .

Модификация полной формы шаблона

Другим подходом к построению кодов полного ЛБШ является использование вместо центрального значения  $f_c$  различных преобразований [26]:

$$ALG(f_c) = \frac{\sum_{j=0}^{P-1} (f_j + f_c)}{P+1}, \tag{7}$$

$$WLG(f_c) = \frac{\sum_{j=0}^{P-1} (f_{c_j} + \alpha f_c)}{P+\alpha}, \tag{8}$$

где  $f_{c_j}$  – равноудалённые соседние пиксели для пикселя  $f_c$ ,  $\alpha \in \mathbb{N}$ . Выражение (8) может быть, например, использовано при вычислении амплитудной составляющей  $CLBP_{-M_{P,R}}$  (5) путём замены  $|f_i - f_c|$  на следующее выражение:

$$m_i = |WLG(f_i) - WLG(f_c)|. \tag{9}$$

Устойчивый локальный бинарный шаблон

Авторы работы [18] предлагают использовать УЛБШ для устранения в процессе формирования кодов шумов, возникающих при небольших отклонениях соседних пикселей от центрального. Для этого вводится некоторое пороговое значение, которое приводит к возникновению дополнительного неизвестного состояния бита кода, как показано далее.

Пусть задано некоторое пороговое значение  $t \in \mathbf{Z}$ , и величина  $\Delta$  обозначает некоторое незаданное (неизвестное) значение из  $\mathbf{B}$ . Для построения неполного кода ЛБШ введём следующую функцию  $\Pi : \mathbf{Z} \rightarrow \mathbf{B} \cup \{\Delta\}$ :

$$\Pi(x) = \begin{cases} 1, & x \geq t, \\ \Delta, & |x| < t, \\ 0, & x \leq -t. \end{cases} \quad (10)$$

В случае, если  $\Pi(x) \in \{0, 1\}$ , то будем считать разницу между значениями пикселей значимой. В противном случае ( $\Pi(x) = \Delta$ ) разница считается неизвестной (малая разница между значениями наиболее подвержена влиянию шума). В этом случае неоднозначный бинарный вектор будем обозначать следующим образом:

$$C(x_c, y_c) = (b_0, b_1, \dots, b_{P-1}), b_i = \Pi(f_i - f_c). \quad (11)$$

Следующим этапом процедуры построения УЛБШ является определение его бинарного вектора таким образом, чтобы шаблон принял стандартный вид. Эта задача решается путём последовательного перебора неизвестных значений в  $C(x_c, y_c)$ . Следует учитывать то, что из одного неоднозначного бинарного вектора можно сформировать несколько ЛБШ стандартного вида (рис. 3). Например, на основе неоднозначного бинарного вектора вида  $1v1v0v0$  могут быть получены следующие стандартные ЛБШ:  $11110000$  и  $11111000$ .

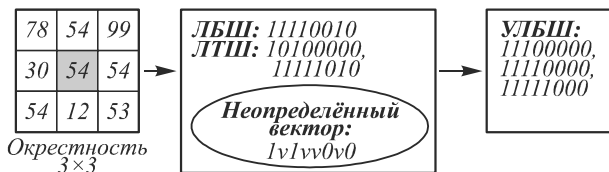


Рис. 3. Формирование УЛБШ ( $t=2$ )

Ввиду того, что неопределённый вектор может породить несколько стандартных ЛБШ, в [26] предлагается алгоритм построения признаков на основе вычисляемых кодов.

Пусть для изображения  $f$  определено скользящее окно обработки с размерами  $M_1 \times M_2$   $W_{k,l}$ ,  $k \in [0, N_1 - M_1]$ ,  $l \in [0, N_2 - M_2]$ , где  $k, l$  – координаты верхнего левого пикселя окна, а  $N_1 \times N_2$  – размеры изображения. Пусть также задано множество всех стандартных ЛБШ  $S_C$ , получаемых из неоднозначного бинарного вектора  $C$ . Тогда алгоритм построения вектора признаков для  $W_{k,l}$  с использованием УЛБШ выглядит следующим образом:

**Шаг 0.** Для всех скользящих окон обработки  $W_{k,l}$  вычислить неоднозначный бинарный вектор  $C(x_c, y_c)$  (11).

**Шаг 1.** Сформировать для каждого  $C(x_c, y_c)$  множество  $S_{C(x_c, y_c)}$ .

**Шаг 2.** Добавить элементы  $t \in S_{C(x_c, y_c)}$  в гистограмму значений стандартных кодов (вектор признаков)  $\mathbf{h}_W(t) \in \mathbf{R}$  следующим образом:

$$\mathbf{h}_W(t) = \begin{cases} \mathbf{h}_W(t) + 1, & S_{C(x_c, y_c)} = \emptyset \\ \mathbf{h}_W(t) + \frac{1}{|S_{C(x_c, y_c)}|}, & \text{иначе} \end{cases}$$

Локальный тернарный шаблон

ЛТШ является расширением ЛБШ и позволяет задавать три различных значения при построении кода, учитывая знак разницы соседних пикселей от центрального.

Пусть  $t \in \mathbf{Z}$  – некоторое пороговое значение, тогда код ЛТШ определяется следующим образом:

$$LTP_{P,R}(x_c, y_c) = \sum_{i=0}^{P-1} \left( I(f_i - f_c \geq t) - I(f_i - f_c \leq -t) \right) \cdot 3^i. \quad (12)$$

ЛТШ также отличается повышенной устойчивостью к искажениям в связи с учётом знака, что отсутствует в ЛБШ. При этом количество возможных кодов ЛТШ при строгом кодовом преобразовании составляет  $3^8 = 6561$  (при  $P=8, R=1$ ). Для снижения количества кодов ЛТШ может быть представлен в виде двух ЛБШ вида (1):

$$LBP_{P,R}^1(x_c, y_c) = \sum_{i=0}^{P-1} I(f_i - f_c \geq t) \cdot 2^i, \quad (13)$$

$$LBP_{P,R}^2(x_c, y_c) = \sum_{i=0}^{P-1} I(f_i - f_c > -t) \cdot 2^i.$$

В таком случае для анализируемого блока строится два кода ЛБШ, которые в дальнейшем используются одновременно в процессе обнаружения дубликатов.

Следует также отметить, что все описанные выше формы ЛБШ справедливы и для ЛТШ и вводятся аналогичным образом.

**2. Бинарные градиентные контура**

БГК (Binary Gradient Contours), предложенные Fernández, A. и коллективом соавторов в работе [19], относятся к классу дескрипторов, основанных на парном сравнении значений пикселей, выбираемых из некоторой окрестности (далее будем использовать окрестность  $3 \times 3$ ) по заранее определённому правилу или маршруту. Очевидно, что количество различных вариантов построения БГК определяется правилами обхода пикселей окрестности. В работе [19] предлагается 3 способа построения БГК.

Первые два способа используют один обход всей окрестности центрального пикселя для генерации кода (обход пикселей для всех способов ведётся по часовой стрелке). В качестве пути обхода пикселей для первого способа применяется последовательность вида  $\{0, 1, 2, 3, 4, 5, 6, 7, 0\}$ , где «0» соответствует индексу верхнего левого пикселя окрестности. Код БГК в этом случае вычисляется следующим образом:

$$BGC^1(x_c, y_c) = \sum_{i=0}^7 I(f_i - f_{(i+1) \bmod 8} \geq 0) \cdot 2^i. \quad (14)$$

Второй способ основан на обходе пикселей по маршруту вида  $\{0, 3, 6, 1, 4, 7, 2, 5, 0\}$ . Код БГК в этом случае выглядит следующим образом:

$$BGC^2(x_c, y_c) = \sum_{i=0}^7 I \left( \begin{matrix} f_{3i \bmod 8} - \\ - f_{3(i+1) \bmod 8} \geq 0 \end{matrix} \right) \cdot 2^i. \quad (15)$$

Оба способа генерируют  $2^8 - 1$  возможных вариантов кодов.

Третий способ генерации кода БГК отличается от предыдущих использованием двух отдельных маршрутов: {1, 3, 7, 5, 1} и {0, 2, 4, 6, 0}. Каждый маршрут позволяет создать  $2^4-1$  различных кодов, а их совместное использование, очевидно, приводит к  $(2^4-1)^2$  различным кодам. Выражение для данного способа генерации кода БГК выглядит следующим образом:

$$\begin{aligned}
 BGC^3(x_c, y_c) = & \\
 = (2^4 - 1) \cdot \sum_{i=0}^3 I(f_{2i} - f_{2(i+1) \bmod 8} \geq 0) \cdot 2^i + & \quad (16) \\
 + \sum_{i=0}^3 I(f_{2i+1} - f_{(2i+3) \bmod 8} \geq 0) \cdot 2^i - 2^4. &
 \end{aligned}$$

**3. Алгоритм обнаружения дубликатов**

В данном пункте будет представлено описание алгоритма обнаружения дубликатов с использованием признакового описания. В основе признакового описания лежит вычисление значений кодов шаблонов. Алгоритм состоит из следующих этапов:

1. Формирование векторов признаков.
2. Поиск близких векторов признаков.
3. Постобработка с целью снижения ошибок обнаружения.

Формирование векторов признаков

Пусть для изображения  $f$  определено окно обработки  $W_{k,l}$  с размерами  $M_1 \times M_2$ , как в предыдущем разделе. Пусть также задано смещение окна  $s \in \mathbf{N}$  (для скользящего окна  $s = 1$ ). Количество окон обработки изображения будет вычисляться следующим образом:

$$L = \left( \left\lfloor \frac{N_1 - M_1}{s} \right\rfloor + 1 \right) \cdot \left( \left\lfloor \frac{N_2 - M_2}{s} \right\rfloor + 1 \right).$$

Пусть далее для каждого  $f(x_c, y_c) \in W_{k,l}$ ,  $x_c \in [k+1, k+M_1-1]$ ,  $y_c \in [l+1, l+M_2-1]$  вычисляется код шаблона  $t(x_c, y_c) \in \mathbf{N}^+$ ,  $\mathbf{N}^+ \equiv \mathbf{N} \cup \{0\}$  по одному из описанных ранее алгоритмов. При этом количество возможных кодов  $Q \in \mathbf{N}$  зависит от используемого шаблона. На основе значений  $t(x_c, y_c)$  для каждого окна обработки вычисляется  $Q$ -мерный вектор признаков  $\mathbf{h} \in \mathbf{R}^Q$  при помощи агрегирующего преобразования (например, гистограммы появления значений кодов для окна обработки). Вычисленные векторы признаков изображения помещаются затем в матрицу  $\mathbf{M}^{L \times Q} = [\mathbf{h}_0, \dots, \mathbf{h}_{L-1}]^T$ . Эта матрица будет использоваться на следующем этапе поиска близких векторов признаков.

Поиск близких векторов признаков

Очевидным способом поиска близких в смысле выбранной меры сравнения векторов признаков является попарное сравнение всех векторов, что является вычислительно неэффективным решением. В целях снижения вычислительной сложности для поиска близких векторов признаков будем использовать одну из разновидностей бинарного дерева поиска – k-d-дерево [27].

Построение и применение дерева будет производиться на основе данных матрицы  $\mathbf{M}^{L \times Q}$ . В ходе про-

граммной реализации была выбрана функция *createns* среды разработки Matlab R2015b. Основными параметрами этой функции являются мера близости векторов признаков («Distance») и количество ближайших соседей, определяемых на этапе применения дерева («K»). В качестве метрики было выбрано евклидово расстояние («euclidean»), а число ближайших соседей  $K = 10$ . Результатом применения k-d-дерева является матрица индексов векторов признаков  $\mathbf{D}^{L \times K}$ , элементы которой  $d_{ij} = \arg(\mathbf{h}_{d_{ij}})$ ,  $d_{ij} \in \mathbf{N}_{[0, L-1]}^+ \setminus \{i\}$  соответствуют индексам векторов признаков, ближайших к вектору признаков  $\mathbf{h}_i$ .

Постобработка

На этапе постобработки будем производить фильтрацию результатов поиска ближайших векторов признаков, тем самым снизив количество ложно обнаруженных и пропущенных дубликатов. В рамках процедуры фильтрации  $\mathbf{D}^{L \times K}$  будем оставлять только такие пары векторов признаков  $(\mathbf{h}_i, \mathbf{h}_j)$ , для которых выполняются следующие два условия:

1.  $\forall i \in [0, L-1], j \in [0, Q-1], \|\mathbf{h}_i - \mathbf{h}_j\| \leq 0,1$ , где  $\|\bullet\|$  – Евклидова норма;
2.  $\forall i \in [0, L-1], j \in [0, Q-1], d(\mathbf{h}_i - \mathbf{h}_{d_{ij}}) \geq 10$  где  $d(\mathbf{h}_i, \mathbf{h}_j)$  – расстояние между векторами признаков  $\mathbf{h}_i$  и  $\mathbf{h}_j$ .

Для каждой удовлетворяющей ограничениям пары векторов признаков будем далее вычислять разницу между координатами положений соответствующих им окон обработки. К дубликатам будем относить такие области анализируемого изображения, которым соответствуют положения окна обработки, расстояния между которыми встречаются наиболее часто среди всех пар векторов признаков  $(\mathbf{h}_i, \mathbf{h}_j)$ .

**4. Экспериментальные исследования**

Для проведения экспериментальных исследований авторами используется стандартный ПК (Intel Core i5-3470 3.2. ГГц, 8 Гб ОЗУ). В качестве объектов исследования были выбраны 10 полутоновых цифровых изображений с размерами 512x512. Авторами была разработана процедура автоматической генерации дубликатов на изображении с возможностью контроля их размеров, количества, а также алгоритма и степени вносимых искажений. Моделировались следующие искажения дубликатов:

1. Линейное контрастирование (мультипликативный коэффициент  $a \in [0, 2, 0, 9]$ , аддитивный  $b \in [5, 30]$ ).
2. Нормальный ( $\sigma < 0,01$ ) и импульсный шум (вероятность возникновения шума –  $p \in [5 \cdot 10^{-2}, 10^{-1}]$ ).
3. Компрессия алгоритмом JPEG (коэффициент качества  $q \in [50, 90]$ ).

Для оценки качества разработанного алгоритма мы использовали три величины, связывающие количество верно обнаруженных дубликатов и значения ошибок первого и второго рода: *Precision*, *Recall* и *F1\_score*. Пусть  $tp$  – количество верно обнаруженных дубликатов,  $fp$  – количество пропущенных дубликатов, а  $fn$  – количество ложно обнаруженных дубликатов. Тогда *Precision* (*Pr*), *Recall* (*Rc*) и *F1\_score* (*F1*) вычисляются следующим образом:

$$Pr = \frac{tp}{tp + fp}, \tag{17}$$

$$Rc = \frac{tp}{tp + fn}, \tag{18}$$

$$F1 = \frac{2 \cdot tp}{2 \cdot tp + fp + fn}. \tag{19}$$

При помощи разработанной процедуры моделирования искажений для каждого из 10 изображений формировалось 120 фальсифицированных изображений (по 40 изображений на каждый вид искажений), которые в дальнейшем обрабатывались алгоритмом обнаружения с использованием всех описанных выше шаблонов. На рис. 4 показаны примеры изображений с различными видами искажений дубликатов.

В проводимых экспериментах будем использовать следующие обозначения типов шаблонов: ЛБШ – *LBP*, стандартный ЛБШ – *ULBP*, ЛТШ – *LTP*, стандартный ЛТШ – *ULTP*, полная форма ЛБШ с модификацией – *Full\_LBP*, устойчивый к повороту ЛБШ – *RIU*, УЛБШ – *RLBP*, БГК (первый способ) – *BGC\_1*, БГК (второй способ) – *BGC\_2*, БГК (третий способ) – *BGC\_3*. В ходе исследований было также получено, что алгоритмы *LBP* и *LTP*, а также *ULBP* и *ULTP* показали близкие результаты во всех экспериментах, поэтому на графиках будем отображать значения показателей качества только для *LTP* и *ULTP*. Параметры используемых шаблонов были фиксированы:  $P = 8, R = 1$ .



Рис. 4. Примеры генерируемых изображений, содержащих дубликаты, искажённые при помощи: а) линейного контрастирования, б) аддитивного импульсного шума, в) компрессии алгоритмом JPEG

Результаты анализа качества разработанного алгоритма в зависимости от параметров линейного контрастирования представлены в табл. 1 и на рис. 5. Видно, что алгоритм с использованием признаков на основе БГК позволяет обнаруживать больший диапазон параметров искажений при линейном контрасти-

ровании. Так, в табл. 1 выделены максимальные значения *Pr* и *Rc* для фиксированных диапазонов мультипликативного коэффициента *a* при линейном контрастировании (аддитивный коэффициент при этом менялся в диапазоне  $b \in [5, 30]$ ).

Табл. 1. Значения Precision-Recall при варьировании параметров линейного контрастирования

		0,2-0,3	0,3-0,4	0,4-0,5	0,5-0,7	0,7-0,9
<b>LTP</b>	<i>Pr</i>	0,62	0,91	0,93	0,92	0,92
	<i>Rc</i>	0,45	0,73	0,90	0,97	0,97
<b>ULTP</b>	<i>Pr</i>	0,53	0,86	0,92	0,94	0,94
	<i>Rc</i>	0,39	0,66	0,82	0,92	0,94
<b>Full_LBP</b>	<i>Pr</i>	0,63	0,87	0,89	0,91	0,94
	<i>Rc</i>	0,46	0,75	0,79	0,84	0,93
<b>RIU</b>	<i>Pr</i>	0,49	0,65	0,83	0,88	0,91
	<i>Rc</i>	0,40	0,54	0,74	0,80	0,88
<b>RLBP</b>	<i>Pr</i>	0,27	0,48	0,79	0,93	0,92
	<i>Rc</i>	0,19	0,33	0,59	0,78	0,84
<b>BGC_1</b>	<i>Pr</i>	<b>0,81</b>	<b>0,90</b>	0,91	0,91	0,89
	<i>Rc</i>	<b>0,57</b>	<b>0,77</b>	0,95	0,98	0,99
<b>BGC_2</b>	<i>Pr</i>	<b>0,75</b>	<b>0,92</b>	0,93	0,94	0,93
	<i>Rc</i>	<b>0,56</b>	<b>0,78</b>	0,94	0,98	0,98
<b>BGC_3</b>	<i>Pr</i>	<b>0,78</b>	<b>0,94</b>	0,95	0,95	0,95
	<i>Rc</i>	<b>0,58</b>	<b>0,81</b>	0,95	0,98	0,98

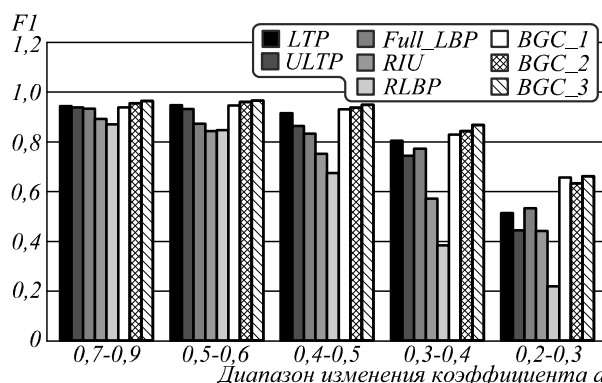


Рис. 5. Зависимость *F1\_score* от параметров линейного контрастирования

При проведении исследований качества обнаружения дубликатов с добавлением шума (импульсного и нормального) было получено также, что использование признаков на основе БГК также позволяет выявлять дубликаты, подвергавшиеся более широкому диапазону их искажения (табл. 2, рис. 6).

В табл. 2 выделены максимальные значения *Pr* и *Rc*, соответствующие признакам на основе БГК, использование которых позволило с более высоким качеством обнаруживать дубликаты с аддитивным белым шумом при  $\sigma^2 \in [10^{-4}, 2 \cdot 10^{-4}]$ . На рис. 6 показаны значения *F1\_score* в зависимости от значения  $\sigma^2$ .

Следует отметить, что при варьировании параметров импульсного шума признаки на основе БГК также показали преимущество перед признаками на основе остальных шаблонов при увеличении вероятности *p*, где  $p \in [5 \cdot 10^{-2}, 10^{-1}]$ . В целях сокращения соответствующие таблица и рисунок не представлены.

Табл. 2. Значения Precision-Recall при варьировании  $\sigma^2$  аддитивного белого шума

		$10^{-5}$	$5 \cdot 10^{-5}$	$10^{-4}$	$2 \cdot 10^{-4}$
<b>LTP</b>	Pr	0,94	0,95	0,94	0,74
	Rc	0,98	0,89	0,76	0,50
<b>ULTP</b>	Pr	0,95	0,89	0,75	0,33
	Rc	0,96	0,79	0,58	0,26
<b>Full_LBP</b>	Pr	0,92	0,85	0,74	0,32
	Rc	0,94	0,80	0,57	0,23
<b>RIU</b>	Pr	0,92	0,83	0,70	0,28
	Rc	0,91	0,77	0,52	0,21
<b>RLBP</b>	Pr	0,95	0,91	0,73	0,33
	Rc	0,97	0,76	0,53	0,24
<b>BGC_1</b>	Pr	0,91	0,94	<b>0,95</b>	<b>0,83</b>
	Rc	0,99	0,89	<b>0,76</b>	<b>0,53</b>
<b>BGC_2</b>	Pr	0,94	0,95	<b>0,96</b>	<b>0,88</b>
	Rc	0,98	0,91	<b>0,81</b>	<b>0,59</b>
<b>BGC_3</b>	Pr	0,95	0,96	<b>0,96</b>	<b>0,78</b>
	Rc	0,98	0,91	<b>0,79</b>	<b>0,53</b>

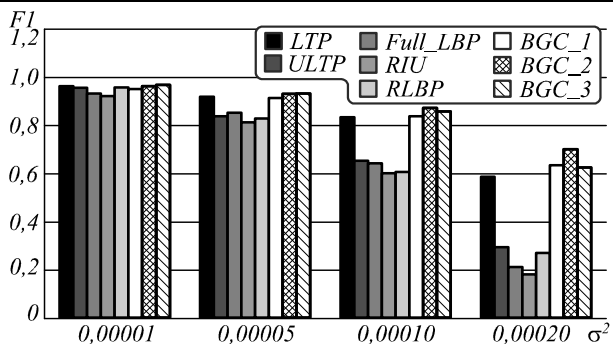


Рис. 6. Зависимость F1\_score от параметров аддитивного белого шума

Исследование зависимости качества обнаружения дубликатов, к которым применялась компрессия алгоритмом JPEG, показало, что признаки на основе БГК (второй и третий способы) позволяют обнаруживать дубликаты, сжатые с качеством  $q \in [50, 90]$ , тогда как использование остальных признаков возможно только при  $q \in [80, 90]$  (табл. 3, рис. 7).

Табл. 3. Значения Precision-Recall при варьировании качества компрессии JPEG

		50	60	70	80	90
<b>LTP</b>	Pr	0,26	0,37	0,53	0,73	0,95
	Rc	0,18	0,23	0,34	0,52	0,75
<b>ULTP</b>	Pr	0,11	0,16	0,27	0,42	0,74
	Rc	0,09	0,12	0,18	0,31	0,54
<b>Full_LBP</b>	Pr	0,29	0,39	0,56	0,76	0,95
	Rc	0,20	0,24	0,37	0,54	0,77
<b>RIU</b>	Pr	0,09	0,15	0,27	0,44	0,75
	Rc	0,08	0,11	0,19	0,32	0,56
<b>RLBP</b>	Pr	0,16	0,23	0,37	0,55	0,85
	Rc	0,12	0,15	0,24	0,39	0,61
<b>BGC_1</b>	Pr	0,27	0,35	0,50	0,66	0,93
	Rc	0,16	0,18	0,29	0,44	0,68
<b>BGC_2</b>	Pr	<b>0,54</b>	<b>0,64</b>	<b>0,73</b>	0,91	0,95
	Rc	<b>0,34</b>	<b>0,40</b>	<b>0,49</b>	0,69	0,81
<b>BGC_3</b>	Pr	<b>0,61</b>	<b>0,71</b>	<b>0,75</b>	0,93	0,96
	Rc	<b>0,43</b>	<b>0,49</b>	<b>0,56</b>	0,75	0,86

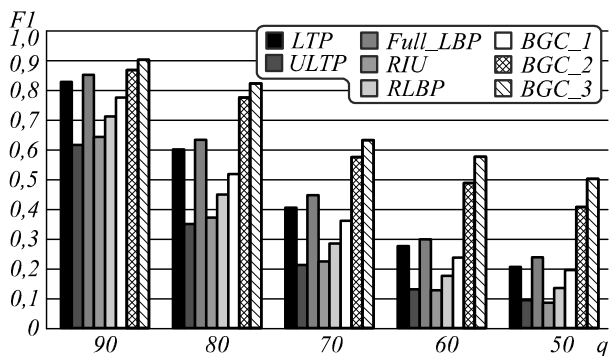


Рис. 7. Зависимость F1\_score от качества компрессии q алгоритмом JPEG

Примеры результатов обнаружения дубликатов на некоторых фальсифицированных изображениях представлены на рис. 8. Из рисунка также видно, что признаки на основе БГК позволяют с более высокой точностью выявлять искажённые дубликаты.



Рис. 8. Фальсифицированное при помощи линейного контрастирования изображение Lena (а), фальсифицированное при помощи линейного контрастирования и JPEG компрессии изображение Lena (б), результат обнаружения дубликатов (в-з)

**Выводы**

В данной работе был предложен алгоритм обнаружения искажённых дубликатов на основе БГК. Проведённые исследования показали повышенную

устойчивость этого вида локальных шаблонов к различным видам искажений дубликатов по сравнению с рядом известных и широко используемых для решения данной задачи ЛБШ и ЛТШ. Признаки на основе БГК позволяют обнаруживать дубликаты на более широком диапазоне параметров вносимых искажений. В дальнейшем планируется повышать точность алгоритма за счёт дополнительных этапов постобработки (фильтрация ошибок обнаружения) и провести исследования алгоритма при использовании комбинированных признаков локальных шаблонов для повышения точности. Также планируется провести исследования геометрически искажённых дубликатов.

### Благодарности

Работы по созданию алгоритма обнаружения дубликатов на основе бинарных градиентных контуров получены в рамках выполнения работ по гранту РФ № 14-31-00014. Результаты экспериментальных исследований получены в рамках выполнения работ по гранту РФФИ № 16-37-00056-мол\_а.

### Литература

1. The Top 20 Valuable Facebook Statistics [Electronical Resource]. – URL: <https://zephoria.com/top-15-valuable-facebook-statistics/> (Request Date 19.01.2015).
2. **Christlein, V.** An Evaluation of Popular Copy-Move Forgery Detection Approaches / V. Christlein, C. Riess, J. Jordan, E. Angelopoulou // IEEE Transactions on information forensics and security. – 2012. – Vol. 7(6). – P. 1841-1854.
3. **Глумов, Н.И.** Поиск дубликатов на цифровых изображениях / Н.И. Глумов, А.В. Кузнецов, В.В. Мясников // Компьютерная оптика. – 2013. – Т. 37, № 3. – С. 360-367.
4. **Кузнецов, А.В.** Алгоритм обнаружения дубликатов на цифровых изображениях с использованием эффективных линейных локальных признаков / А.В. Кузнецов, В.В. Мясников // Компьютерная оптика. – 2013. – Т. 37, № 4. – С. 489-495.
5. **Kuznetsov, A.V.** A fast plain copy-move detection algorithm based on structural pattern and 2D Rabin-Karp rolling hash / A.V. Kuznetsov, V.V. Myasnikov // Lecture Notes in Computer Science. – 2014. – Vol. 8814. – P. 461-468.
6. **Mahdian, B.** Detection of Copy-Move Forgery using a Method Based on Blur Moment Invariants / B. Mahdian, S. Saic // Forensic Science International. – 2007. – Vol. 171(2). – P. 180-189.
7. **Ryu, S.** Detection of Copy-Rotate-Move Forgery using Zernike Moments / S. Ryu, M. Lee, H. Lee // Information Hiding Conference. – 2010. – P. 51-65.
8. **Popescu, A.** Exposing digital forgeries by detecting duplicated image regions [Electronical Resource] / A. Popescu, H. Farid. – 2004. – URL: <http://www.ists.dartmouth.edu/library/102.pdf> (Request Date 20.01.2015).
9. **Kang, X.** Identifying Tampered Regions Using Singular Value Decomposition in Digital Image Forensics / X. Kang, S. Wei // International Conference on Computer Science and Software Engineering. – 2008. – Vol. 3. – P. 926-930.
10. **Luo, W.** Robust Detection of Region-Duplication Forgery in Digital Images / W. Luo, J. Huang, G. Qiu // International Conference on Pattern Recognition. – 2006. – Vol. 4. – P. 746-749.
11. **Bravo-Solorio, S.** Exposing Duplicated Regions Affected by Reflection, Rotation and Scaling / S. Bravo-Solorio, A.K. Nandi // International Conference on Acoustics, Speech and Signal Processing. – 2011. – P. 1880-1883.
12. **Fridrich, J.** Detection of copy-move forgery in digital images [Electronical Resource] / J. Fridrich, D. Soukal, J. Lukas. – 2003. – URL: <http://www.ws.binghamton.edu/fridrich/Research/copy-move.pdf> (Request Date 20.01.2015).
13. **Bayram, S.** An efficient and robust method for detecting copy-move forgery / S. Bayram, H. Sencar, N. Memon // IEEE International Conference on Acoustics, Speech, and Signal Processing. – 2009. – P. 1053-1056.
14. **Huang, H.** Detection of Copy-Move Forgery in Digital Images Using SIFT Algorithm / H. Huang, W. Guo, Y. Zhang // Pacific-Asia Workshop on Computational Intelligence and Industrial Application. – 2008. – Vol. 2. – P. 272-276.
15. **Shivakumar, B.L.** Detection of Region Duplication Forgery in Digital Images Using SURF / B.L. Shivakumar, S. Baboo // International Journal of Computer Science Issues. – 2011. – Vol. 8(4). – P. 199-205.
16. **Li, L.** An Efficient Scheme for Detection Copy-move Forged Images by Local Binary Patterns / L. Li, S. Li, H. Zhu // Journal of Information Hiding and Multimedia Signal Processing. – 2013. – Vol. 4(1). – P. 46-56.
17. **Davarzani, R.** Copy-move forgery detection using multiresolution local binary patterns / R. Davarzani, K. Yaghmaie, S. Mozaffari, M. Tapak // Forensic Science International. – 2013. – Vol. 231(1-3). – P. 61-72.
18. **Ren, J.** Noise-Resistant Local Binary Pattern With an Embedded Error-Correction Mechanism / J. Ren, X. Jiang, J. Yuan // IEEE Transactions on Image Processing. – 2013. – Vol. 22(10). – P. 4049-4060.
19. **Fernández, A.** Image classification with binary gradient contours / A. Fernández, M.X. Álvarez, F. Bianconi // Optics and Lasers in Engineering. – 2011. – Vol. 49, Issue. 9-10. – P. 1177-1184.
20. **Wang, L.** Texture classification using texture spectrum / L. Wang, D.-C. He // Pattern Recognition. – 1990. – Vol. 23(8). – P. 905-910.
21. **Ojala, T.** A comparative study of texture measures with classification based on featured distribution / T. Ojala, M. Pietikinen, D. Harwood // Pattern Recognition. – 1996. – Vol. 29(1). – P. 51-59.
22. **Ojala, T.** Multiresolution grayscale and rotation invariant texture classification with local binary patterns / T. Ojala, M. Pietikinen, T. Menp // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2002. – Vol. 24(7). – P. 971-987.
23. **Мясников, В.В.** Локальное порядковое преобразование цифровых изображений / В.В. Мясников // Компьютерная оптика. – 2015. – Т. 39, № 3. – С. 397-405.
24. **Arasteh, S.** Color and texture image segmentation using uniform local binary patterns / S. Arasteh, C.-C. Hung // Machine Graphics and Vision. – 2006. – Vol. 15(3-4). – P. 265-274.
25. **Guo, Z.H.** A completed modeling of local binary pattern operator for texture classification / Z.H. Guo, D. Zhang // IEEE Transactions on Image Processing. – 2010. – Vol. 19(6). – P. 1657-1663.
26. **Zhao, Y.** Completed robust local binary pattern for texture classification / Y. Zhao, W. Jia, R.X. Hu, H. Min // Neurocomputing. – 2013. – Vol. 106. – P. 68-76.
27. **Bentley, J.L.** Multidimensional binary search trees used for associative searching / J.L. Bentley // Communications of the ACM. – 1975. – Vol. 18(9). – P. 509-517.



**Сведения об авторах**

**Кузнецов Андрей Владимирович**, родился в 1987 году. В 2010 году окончил Самарский государственный аэрокосмический университет (СГАУ) с отличием по специальности «Прикладная математика и информатика». В 2010 поступил в аспирантуру СГАУ, в 2013 г. защитил диссертацию на соискание степени кандидата технических наук. В настоящее время работает старшим научным сотрудником НИЛ-97 СГАУ и научным сотрудником в Институте систем обработки изображений РАН. Круг научных интересов включает обработку и анализ изображений, распознавание образов, обнаружение локальных изменений на изображениях, геоинформатику. Имеет 37 публикаций, в том числе 18 научных статей и 1 монографию (в соавторстве). Страница в интернете: <http://nil97.ssau.ru/employee/detail.php?ID=35>. E-mail: [kuznetsoff.andrey@gmail.com](mailto:kuznetsoff.andrey@gmail.com).

Сведения об авторе **Мясников Владислав Валерьевич** см. стр. 281 этого номера.

*Поступила в редакцию 19 февраля 2016 г.  
Окончательный вариант – 14 марта 2016 г.*

**A COPY-MOVE DETECTION ALGORITHM BASED ON BINARY GRADIENT CONTOURS**

*A.V. Kuznetsov<sup>1</sup>, V.V. Myasnikov<sup>1,2</sup>*

*<sup>1</sup> Samara State Aerospace University, Samara, Russia,*

*<sup>2</sup> Image Processing Systems Institute, Russian Academy of Sciences, Samara, Russia*

**Abstract**

Copy-move is one of the most obvious ways of deliberate distortion of digital images in order to conceal the information contained in them. The process of duplicate embedding consists in copying an image fragment and pasting it within the same image. Prior to pasting, the fragment can be distorted using transformations such as contrast enhancement, noise adding, scaling, rotation, and combinations thereof. Existing approaches to copy-move forgery detection are based on calculating feature vectors for overlapping blocks of an image and then using these vectors to find the closest regions in Euclidean space. In this paper, we propose features based on binary gradient contours, which are resistant to contrast enhancement, additive noise and JPEG compression. We also present results of conducted experiments for demonstrating the proposed algorithm effectiveness for a range of distortion parameters. The research also involves comparing features based on binary gradient contours with features based on various forms of local binary patterns.

**Keywords:** copy-move detection, distorted duplicate, local binary pattern, binary gradient contours, feature vector, k-d tree.

**Citation:** Kuznetsov AV, Myasnikov VV. A copy-move detection algorithm based on binary gradient contours. *Computer Optics* 2016; 40(2): 284-93. DOI: 10.18287/2412-6179-2016-40-2-284-293.

**Acknowledgements:** The work on development of the copy-move detection algorithm based on binary gradient contours was carried out while working on the RSF grant No. 14-31-00014. Experimental results were obtained while working on the RFBR grant No. 16-37-00056-мол\_а.

**References**

- [1] The Top 20 Valuable Facebook Statistics. Source: (<https://zephoria.com/top-15-valuable-facebook-statistics>).
- [2] Christlein V, Riess C, Jordan J, Angelopoulou E. An Evaluation of Popular Copy-Move Forgery Detection Approaches. *IEEE Transactions on information forensics and security* 2012; 7(6): 1841-1854.
- [3] Glumov NI, Kuznetsov AV, Myasnikov VV. The algorithm for copy-move detection on digital images. *Computer Optics* 2013; 37(3): 360-367.
- [4] Kuznetsov AV, Myasnikov VV. Efficient linear local features based copy-move detection algorithm. *Computer Optics* 2013; 37(4): 489-495.
- [5] Kuznetsov AV, Myasnikov VV. A fast plain copy-move detection algorithm based on structural pattern and 2D Rabin-Karp rolling hash. *LNCS* 2014; 8814: 461-468.
- [6] Mahdian B, Saic S. Detection of Copy-Move Forgery using a Method Based on Blur Moment Invariants. *Forensic Science International* 2007; 171(2): 180-189.
- [7] Ryu S, Lee M, Lee H. Detection of Copy-Rotate-Move Forgery using Zemike Moments. *Information Hiding Conference* 2010: 51-65.
- [8] Popescu A, Farid H. Exposing digital forgeries by detecting duplicated image regions. Source: (<http://www.ists.dartmouth.edu/library/102.pdf>).
- [9] Kang X, Wei S. Identifying Tampered Regions Using Singular Value Decomposition in Digital Image Forensics. *International Conference on Computer Science and Software Engineering* 2008; 3: 926-930.
- [10] Luo W, Huang J, Qiu G. Robust Detection of Region-Duplication Forgery in Digital Images. *International Conference on Pattern Recognition* 2006; 4: 746-749.
- [11] Bravo-Solorio S, Nandi AK. Exposing Duplicated Regions Affected by Reflection, Rotation and Scaling. *International Conference on Acoustics, Speech and Signal Processing* 2011: 1880-1883.

- [12] Fridrich J, Soukal D, Lukas J. Detection of copy-move forgery in digital images. Source: <http://www.ws.binghamton.edu/fridrich/Research/copymove.pdf>.
- [13] Bayram S, Sencar H, Memon H. An efficient and robust method for detecting copy-move forgery. IEEE International Conference on Acoustics, Speech, and Signal Processing 2009: 1053-1056.
- [14] Huang H, Guo W, Zhang Y. Detection of Copy-Move Forgery in Digital Images Using SIFT Algorithm. Pacific-Asia Workshop on Computational Intelligence and Industrial Application 2008; 2: 272-276.
- [15] Shivakumar BL, Baboo S. Detection of Region Duplication Forgery in Digital Images Using SURF. International Journal of Computer Science 2011; 8(4): 199-205.
- [16] Li L, Li S, Zhu H. An Efficient Scheme for Detection Copy-move Forged Images by Local Binary Patterns. Journal of Information Hiding and Multimedia Signal Processing 2013; 4(1): 46-56.
- [17] Davarzani R, Yaghmaie K, Mozaffari S, Tapak M. Copy-move forgery detection using multi-resolution local binary patterns. Forensic Science International 2013; 231(1-3): 61-72.
- [18] Ren J, Jiang X, Yuan J. Noise-Resistant Local Binary Pattern with an Embedded Error-Correction Mechanism. IEEE Transactions on Image Processing 2013; 22(10): 4049-4060.
- [19] Fernández A, Álvarez MX, Bianconi F. Image classification with binary gradient contours. Opt Lasers Eng 2011; 49(9-10): 1177-1184.
- [20] Wang L, He D-C. Texture classification using texture spectrum. Pattern Recognition 1990; 23(8): 905-910.
- [21] Ojala T, Pietikinen M, Harwood D. A comparative study of texture measures with classification based on featured distribution. Pattern Recognition 1996; 29(1): 51-59.
- [22] Ojala T, Pietikinen M, Menp T. Multiresolution grayscale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence 2002; 24(7): 971-987.
- [23] Myasnikov VV. A local order transform of digital images. Computer Optics 2015; 39(3): 397-405.
- [24] Arasteh S, Hung C-C. Color and texture image segmentation using uniform local binary patterns. Machine Graphics and Vision 2006; 15(3-4): 265-274.
- [25] Guo ZH, Zhang D. A completed modeling of local binary pattern operator for texture classification. IEEE Transactions on Image Processing 2010; 19(6): 1657-1663.
- [26] Zhao Y, Jia W, Hu RX, Min H. Completed robust local binary pattern for texture classification. Neuro-computing 2013; 106: 68-76.
- [27] Bentley JL. Multidimensional binary search trees used for associative searching. Communications of the ACM 1975; 18(9): 509-517.

---

#### *Authors' information*

**Andrey Vladimirovich Kuznetsov** (b. 1987) graduated with honours (2010) from Samara State Aerospace University (SSAU), majoring in Applied Mathematics and Informatics. He studied as a post-graduate student at SSAU from 2010 and received his PhD in Technical Sciences in 2013. Nowadays he is a senior researcher at the Remote Sensing Data Analysis laboratory at SSAU and a researcher at the Image Processing Systems Institute of the Russian Academy of Sciences (IPSI RAS). His research interests are currently focused on image processing and analysis, pattern recognition, digital image forgery detection, geoinformatics. He has 37 publications, including 18 scientific papers and 1 monograph (with coauthors). Web-page: <http://ni197.ssau.ru/employee/detail.php?ID=35> . E-mail: [kuznetsoff.andrey@gmail.com](mailto:kuznetsoff.andrey@gmail.com) .

The information about author **Vladislav Valerievich Myasnikov** you can find on page 283 of this issue.

---

*Received February 19, 2016. The final version – March 14, 2016.*

---