

ЛОКАЛЬНЫЕ ШАБЛОНЫ В ЗАДАЧЕ ОБНАРУЖЕНИЯ ДУБЛИКАТОВ

Н.И. Евдокимова¹, А.В. Кузнецов^{1,2}

¹Самарский национальный исследовательский университет имени академика С.П. Королева, Самара, Россия,

²Институт систем обработки изображений РАН – филиал ФНИЦ «Кристаллография и фотоника» РАН, Самара, Россия

Аннотация

Встраивание дубликатов является одним из часто применяемых способов подделки изображений, при котором некоторый фрагмент изображения копируется в другую его часть с целью сокрытия какой-либо информативной части. Целью алгоритма обнаружения дубликатов является выявление областей дубликатов на изображении. В основе алгоритма обнаружения дубликатов лежит вычисление признаков в скользящем окне или окне с перекрытиями. Данная работа посвящена сравнению алгоритмов обнаружения дубликатов с использованием признаков на основе локальных бинарных шаблонов, локальных тернарных шаблонов, локальных дифференцирующих шаблонов и некоторых их расширений. В части исследований приводятся результаты анализа точности обнаружения дубликатов при помощи рассматриваемых методов. Отличительной особенностью используемых признаков является устойчивость к вносимым в дубликат искажениям, таким как линейное контрастирование, импульсный шум и низкая вычислительная сложность.

Ключевые слова: дубликат, искажение, локальный бинарный шаблон, локальный тернарный шаблон, локальный дифференцирующий шаблон.

Цитирование: Евдокимова, Н.И. Локальные шаблоны в задаче обнаружения дубликатов / Н.И. Евдокимова, А.В. Кузнецов // Компьютерная оптика. – 2017. – Т. 41, № 1. – С. 79-87. – DOI: 10.18287/2412-6179-2017-41-1-79-87.

Введение

В настоящее время цифровые изображения и видео являются одними из основных средств передачи информации. Например, они используются для подтверждения каких-либо событий в СМИ. Широкая доступность программного обеспечения привела к тому, что осуществить подделку изображения можно очень просто. Для обеспечения безопасности данных исследователи на протяжении последних 10–15 лет разрабатывают алгоритмы обнаружения искусственных искажений на цифровых изображениях [1–3]. С помощью таких методов можно определять области изображения, подвергавшиеся изменениям, а также характеристики вносимых злоумышленниками искажений.

Встраивание дубликатов является одним из самых часто используемых методов подделки изображений в силу простоты его использования. Процесс встраивания состоит из трёх последовательных этапов: копирование некоторой области изображения, искажение этой области и встраивание её в той части этого же изображения, которую необходимо скрыть. Несмотря на большое количество существующих решений, многие из них не являются вычислительно эффективными. Именно поэтому на сегодняшний день разработка методов, позволяющих быстро и с высокой точностью обнаруживать изменённые области, является актуальной.

Работа построена следующим образом. В первом параграфе рассмотрены основные сведения о локальном бинарном шаблоне (ЛБШ), локальном тернарном шаблоне (ЛТШ), некоторых их разновидностях и локальном дифференцирующем шаблоне (ЛДШ). Вторым параграф посвящён описанию алгоритма обнаружения дубликатов. Третий параграф содержит в себе

результаты экспериментальных исследований на наборе изображений с дубликатами.

1. Локальные шаблоны

Локальный бинарный шаблон

Идея ЛБШ принадлежит Wang [4]. В базовой версии для построения ЛБШ используется окно 3×3 . Восемь соседних значений сравниваются со значением центрального пикселя, и в зависимости от результата сравнения центральному пикселю ставится в соответствие восьмибитное число – код ЛБШ (рис. 1).

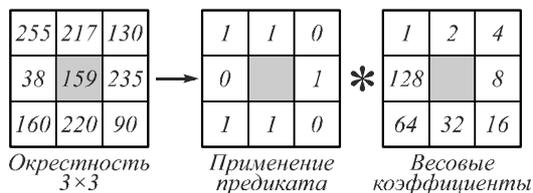


Рис. 1. Базовая версия оператора ЛБШ

Определим изображение как функцию яркости пикселя $f(x, y)$. Код ЛБШ вычисляется в каждом пикселе с использованием значений пикселей в круговой окрестности $f_i = f(x_i, y_i)$, $i = \overline{1, N}$, где N – количество пикселей, попадающих в окрестность, ограниченную окружностью радиуса R с центром в пикселе f_0 (рис. 2), следующим образом:

$$LBP(N, R) = \sum_{i=1}^N I_1(f_i - f_0) \cdot 2^i, \quad (1)$$

где I_1 – индикатор события, определяемый следующим образом:

$$I_1(m) = \begin{cases} 1, & m > 0 \\ 0, & m \leq 0 \end{cases}. \quad (2)$$

Следует также отметить, что оператор ЛБШ очень эффективен благодаря своей низкой вычислительной

сложности. Главным недостатком данного вида шаблонов является чувствительность к шумовым искажениям [5].

f_1	f_2	f_3
f_8	f_0	f_4
f_7	f_6	f_5

Рис. 2. Пример окрестности 3×3 вокруг пикселя f_0 при $R=1$

Центрально-симметричный локальный бинарный шаблон

Центрально-симметричный локальный бинарный шаблон (ЦС-ЛБШ) был представлен Heikkilä в [6]. Оператор ЦС-ЛБШ похож на ЛБШ, но использует другую схему сравнения пикселей в окрестности центрального пикселя. В то время как в ЛБШ центральный пиксель играет ключевую роль, метод ЦС-ЛБШ рассматривает только центрально-симметричные пары значений пикселей. В соответствии с рис. 2 это (f_1, f_5) , (f_2, f_6) , (f_3, f_7) и (f_4, f_8) .

ЦС-ЛБШ может быть определен следующим образом:

$$CS-LBP(N, R, T) = \sum_{i=1}^{N/2} I_2(f_i - f_{i+(N/2)}) \cdot 2^i, \quad (3)$$

где T – некоторый порог, позволяющий достичь устойчивости на однородных областях изображения, а I_2 – индикатор события, определяемый следующим образом:

$$I_2(m) = \begin{cases} 1, & m > T \\ 0, & m \leq T \end{cases} \quad (4)$$

Данная модификация базового оператора ЛБШ позволяет сократить количество получаемых кодов: для окрестности 3×3 оператор ЛБШ позволяет сформировать 256 различных кодов, а оператор ЦС-ЛБШ – 16, что позволяет сделать алгоритм вычислительно более эффективным. Кроме того, этот оператор имеет ряд преимуществ, таких как невосприимчивость к изменениям освещенности и устойчивость на однородных участках изображения [6]. Основными недостатками оператора ЦС-ЛБШ является неинвариантность к шумам, а также необходимость жесткого задания порога T .

Улучшенный локальный бинарный шаблон

Базовая версия ЛБШ учитывает только значения соседних пикселей центрального пикселя при формировании бинарного кода. Таким образом, происходит потеря информации, заключенной в самом центральном пикселе. Как упоминалось ранее, для окрестности 3×3 оператор ЛБШ формирует 256 различных кодов из $2^9 = 512$, которые могли бы быть сформированы с учетом центрального пикселя. Метод, учитывающий значение центрального пикселя при построении бинарного кода, был представлен Jin с соавторами в [7]. Шаблон, используемый в данном методе, носит название улучшенного локального бинарного шаблона (УЛБШ).

В связи с высокой корреляцией между соседними пикселями будем считать, что центральный пиксель содержит в себе больше информации, чем его соседи, так он должен получить наибольший вес при формировании кода. С учетом данной идеи УЛБШ определяется следующим образом:

$$ILBP(N, R, T_{mean}) = \sum_{i=0}^N I_3(f_i - T_{mean}) \cdot 2^i, \quad (5)$$

где I_3 – индикатор события, определяемый следующим образом:

$$I_3(m) = \begin{cases} 1, & m > 0 \\ 0, & m \leq 0 \end{cases} \quad (6)$$

а T_{mean} – среднее значение пикселей по всей окрестности:

$$T_{mean} = \frac{1}{N+1} \cdot \sum_{i=0}^N f_i. \quad (7)$$

УЛБШ инвариантен к изменениям освещенности за счет того, что позволяет более детально описывать артефакты на изображении и информацию о текстуре.

Медианный бинарный шаблон

Пусть в окрестность $M \times M$ попало N пикселей. Упорядочим их по возрастанию и перенумеруем, тем самым получая вариационный ряд:

$$f_1 \leq f_2 \leq \dots \leq f_N. \quad (8)$$

Назовем медианой середину данного ряда, то есть значение $f_{(N+1)/2}$.

Медианный бинарный шаблон (МБШ) [8] имеет много общего с улучшенным локальным бинарным шаблоном. Единственное отличие состоит в том, что вместо среднего значения пикселей по окрестности МБШ использует медиану рассматриваемой окрестности.

МБШ определяется следующим образом:

$$MBP(N, R, T_{median}) = \sum_{i=0}^N I_3(f_i - T_{median}) \cdot 2^i, \quad (9)$$

где T_{median} – медиана окрестности.

Локальный тернарный шаблон

Метод локальных тернарных шаблонов (ЛТШ) был предложен Tan и Triggs [9] для решения проблемы чувствительности к шуму в близких к однородным областям изображения. Данный метод позволяет задавать три различных значения при построении кода, учитывая знак разницы центрального пикселя с соседними.

Тройное кодирование достигается путем введения порогового значения T в формулу базового ЛБШ (2):

$$I_4(m) = \begin{cases} 1, & m \geq T \\ 0, & |m| \leq T \\ -1, & m \leq -T \end{cases}, \quad (10)$$

тогда код ЛТШ определяется следующим образом:

$$LTP(N, R) = \sum_{i=1}^N I_4(f_i - f_0) \times 3^i. \quad (11)$$

Количество возможных кодов ЛТШ составляет $3^8 = 6561$. Такое количество кодов приведет к повы-

шению вычислительной сложности в случае необходимости подсчета частоты их появления (например, при помощи гистограммы). Для устранения этого недостатка ЛТШ может быть представлен в виде двух ЛБШ: положительного и отрицательного, как показано на рис. 3.

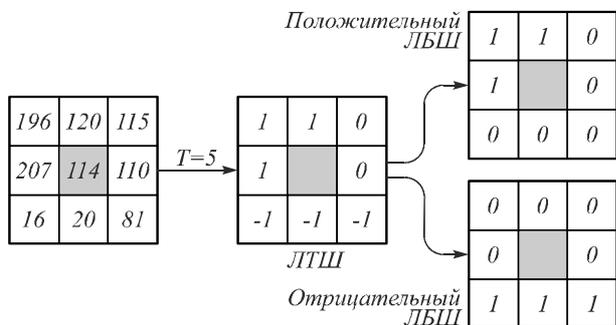


Рис. 3. Пример разделения ЛТШ на положительный ЛБШ и отрицательный ЛБШ

Улучшенный локальный тернарный шаблон

Улучшенный локальный тернарный шаблон (УЛТШ) был введен как расширение базового ЛТШ в [10] Nanni и др. УЛТШ, подобно УЛБШ, использует значение центрального пикселя при формировании кода, а каждый пиксель из его окрестности принимается во внимание с учетом всех остальных в данной окрестности. Как и в базовой версии ЛТШ, УЛТШ может быть представлен в виде двух частей: положительной и отрицательной:

$$ILTP_-(N, R, T_{mean}) = \sum_{i=0}^N I_3(T_{mean} - f_i - T) \cdot 2^i, \quad (12)$$

$$ILTP_+(N, R, T_{mean}) = \sum_{i=0}^N I_3(f_i - T_{mean} - T) \cdot 2^i, \quad (13)$$

где T_{mean} определяется формулой (7), а T – некоторое пороговое значение, аналогичное используемому в (10).

Итоговое значение кода формируется путём объединения положительной и отрицательной частей.

Локальный дифференцирующий шаблон

Код ЛБШ представляет из себя результат применения производной первого порядка к окрестности центрального пикселя, как было показано в (1). В целях более информативного описания изображения в [11] был предложен оператор ЛДШ, который позволяет кодировать информацию, получаемую в результате вычисления производных более высокого порядка. Используемая в данном методе производная n -го порядка по различным направлениям вычисляется на основе функции бинарного кодирования.

Производную первого порядка по направлениям $0^\circ, 45^\circ, 90^\circ$ и 135° будем обозначать как $f'_\alpha(x_0, y_0)$, где $\alpha = 0^\circ, 45^\circ, 90^\circ, 135^\circ$.

Производные первого порядка в точке (x_0, y_0) могут быть записаны в следующем виде:

$$\begin{aligned} f'_0(x_0, y_0) &= f_0 - f_4, & f'_{45}(x_0, y_0) &= f_0 - f_3, \\ f'_{90}(x_0, y_0) &= f_0 - f_2, & f'_{135}(x_0, y_0) &= f_0 - f_1. \end{aligned} \quad (14)$$

ЛДШ второго порядка по направлениям a в точке (x_0, y_0) определяется следующим выражением:

$$LDP^2_\alpha(x_0, y_0) = \left\{ \begin{aligned} &I_2(f'_\alpha(x_0, y_0), f'_\alpha(x_1, y_1)), \\ &I_2(f'_\alpha(x_0, y_0), f'_\alpha(x_2, y_2)), \\ &\dots \\ &I_2(f'_\alpha(x_0, y_0), f'_\alpha(x_8, y_8)) \end{aligned} \right\}, \quad (15)$$

где I_2 – индикатор события, определяемый следующим образом:

$$I_2(m, n) = \begin{cases} 0, & m \cdot n < 0 \\ 1, & m \cdot n \geq 0 \end{cases}. \quad (16)$$

Окончательно ЛДШ второго порядка определяется как объединение четырёх восьмибитных ЛДШ, вычисленных по заданным направлениям:

$$LDP^2(x, y) = \{LDP^2_\alpha(x, y) \mid \alpha = 0^\circ, 45^\circ, 90^\circ, 135^\circ\}. \quad (17)$$

Как видно из приведённых выше уравнений, оператор ЛДШ присваивает код каждому пикселю изображения путём сравнения двух производных по направлению в двух соседних пикселях. Далее следует объединение полученных кодов в единую последовательность из 32 бит. Сравнения производных по направлению выполняются с помощью 16 примитивных шаблонов [11] (рис. 4), характеризующих отличительные пространственные отношения в локальной области. В отличие от ЛБШ, ЛДШ второго порядка позволяет кодировать изменение значения производной по направлению, которое является характеристикой локальной области изображения.

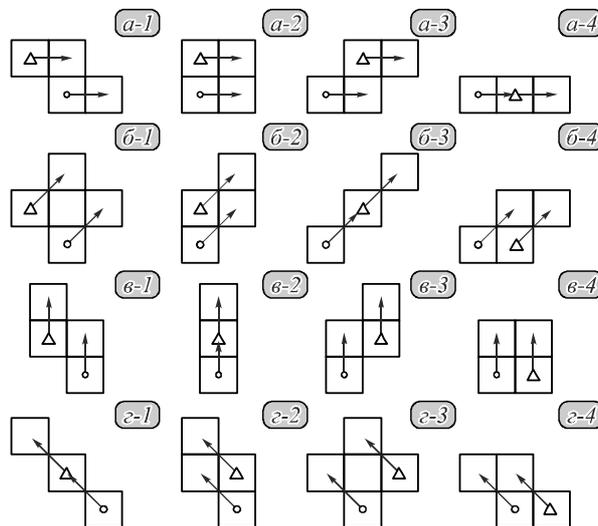


Рис. 4. Примитивные шаблоны, используемые в ЛДШ

Рис. 5 иллюстрирует типы локальных переходов в примитивных шаблонах, используемых в ЛДШ. Переходы представляют из себя бинарные значения. Каждый из 16 примитивных шаблонов на рис. 4 принадлежит одной из двух групп: трёхпиксельные (3-point) или четырёхпиксельные (4-point). Трёхпиксельному шаблону в соответствии с (16) присваивается значение 0, если он является монотонно возраста-

ющим или убывающим (рис. 5б), в противном случае шаблону присваивается значение 1 (рис. 5а). Аналогично четырёхпиксельному шаблону присваивается значение 1, если он представляет из себя «поворот градиента» (рис. 5б), и значение 0, если он монотонно возрастает или убывает (рис. 5з). Таким образом, при помощи локальных переходов на основе примитивных шаблонов формируется бинарный код области изображения.

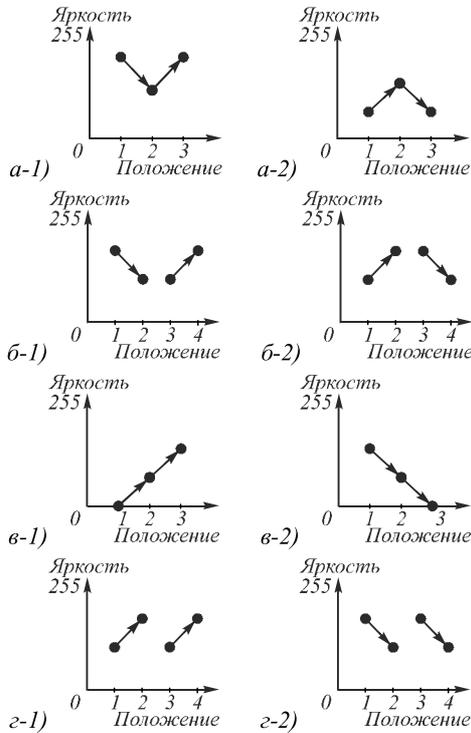


Рис. 5. Типы локальных переходов, используемых для кодирования примитивных шаблонов ЛДШ

В качестве примера рассмотрим некоторую область изображения, представленную на рис. 6.

122	125	123	125	121
126	127	129	121	125
122	123	124	128	122
123	122	123	122	129
121	122	123	122	121

Рис. 6. Область изображения, $f_0 = 124$

Для вычисления ЛДШ второго порядка по направлению $\alpha = 0^\circ$ в точке f_0 четыре шаблона из рис. 4а применяются к изображению путём совмещения * и Δ с положением f_0 последовательно. При применении шаблона а-1 (рис. 4а) путём совмещения * с f_0 обе производные по направлению монотонно возрастают, как это показано на рис. 5з-1. Таким образом, примитивному шаблону соответствует значение 0. Аналогично применение шаблонов из а-2, а-3 и а-4 (рис. 4а) путём совмещения * с f_0 соответствует случаям на рис. 5б-1, рис. 5з-1 и рис. 5а-2 соответственно. В результате вычисляются следующие три бита ЛДШ 101. Повторяя описанную выше процедуру

ру с теми же четырьмя шаблонами, но на этот раз совмещая Δ с f_0 , мы получаем последовательность бит 0100 для последних четырёх битов. В результате формируется код 01010100 для направления $\alpha = 0^\circ$. Очевидным образом формируются коды примитивных шаблонов по направлениям $\alpha = 45^\circ, 90^\circ, 135^\circ$, представленных на рис. 4б-4г соответственно. В итоге получаем 32-битный код:

$$LDP^2(f_0) = 0101010000101111101000011000110,$$

который генерируется путём объединения четырёх 8-битных ЛДШ, вычисленных по направлениям $\alpha = 0^\circ, \alpha = 0^\circ, 45^\circ, 90^\circ, 135^\circ$ в соответствии с выражением (17).

Локальный дифференцирующий шаблон n-го порядка

Вычисляя производные n-го порядка по направлениям $\alpha = 0^\circ, 45^\circ, 90^\circ, 135^\circ$ для пикселя f_0 , можно получить ЛДШ n-го порядка, который определяется следующим образом:

$$LDP^n_\alpha(x_0, y_0) = \left\{ \begin{matrix} I_2(f_\alpha^{n-1}(x_0, y_0), f_\alpha^{n-1}(x_1, y_1)), \\ I_2(f_\alpha^{n-1}(x_0, y_0), f_\alpha^{n-1}(x_2, y_2)), \\ \dots \\ I_2(f_\alpha^{n-1}(x_0, y_0), f_\alpha^{n-1}(x_8, y_8)) \end{matrix} \right\}, \quad (18)$$

где $f_\alpha^{n-1}(x_0, y_0)$ – производная (n-1)-го порядка по направлению α в точке (x_0, y_0) , которая описывает изменения градиента (n-1)-го порядка, предоставляя дополнительную информацию о локальной окрестности. Окончательно ЛДШ n-го порядка определяется следующим выражением:

$$LDP^n(x, y) = \{LDP^n_\alpha(x, y) | \alpha = 0^\circ, 45^\circ, 90^\circ, 135^\circ\}. \quad (19)$$

Таким образом, ЛДШ также строится на основе производных по четырём направлениям.

Локальные шаблоны высокого порядка позволяют более детально описывать локальную окрестность изображения, чем локальный шаблон первого порядка, который используется в ЛБШ. Тем не менее, они являются более чувствительными к шуму в случае роста порядка производной. В работе [11] авторы экспериментально получили, что чувствительность к шуму у ЛДШ повышается при $n \geq 4$. Учитывая это ограничение, в дальнейшем будем использовать только ЛДШ второго и третьего порядков (ЛДШ-2 и ЛДШ-3).

2. Алгоритм обнаружения дубликатов

В рамках данной работы будем использовать алгоритм, предложенный в работах А.В. Кузнецова, В.В. Мясникова и Н.И. Глумова [12–13].

Пусть исходное изображение I, содержащее в себе дублирующиеся фрагменты, имеет размер $N_1 \times N_2$, тогда матрицей хэш-значений назовем матрицу M размерности $N_1 \times N_2$, каждый элемент которой является значением перцептивной хэш-функции кода локального шаблона в пикселе исходного изображения, имеющем те же самые координаты.

Алгоритм может быть представлен в виде следующих основных этапов:

1. В режиме скользящего окна, имеющего размер $k \times k$, производится:

1.1. Вычисление значения кода локального шаблона (ЛБШ, ЦС-ЛБШ, УЛБШ, МБШ, ЛТШ, УЛТШ и ЛДШ).

1.2. Вычисление значения перцептивной хэш-функции (отображение значений матрицы кодов в некоторую целочисленную неотрицательную величину или хэш-значение) [13].

1.3. Помещение полученного значения хэш-функции в соответствующий элемент матрицы хэш-значений.

2. Построение гистограммы хэш-значений на основе матрицы хэш-значений.

3. Анализ гистограммы.

4. Поиск соответствующих элементов в матрице хэш-значений.

5. Формирование выходного бинарного изображения, содержащего выделенные дублирующиеся области.

6. Обработка полученного бинарного изображения фильтром эрозии для уменьшения количества ошибок второго рода. Размеры окна эрозии выбираются экспериментальным путем.

7. Наложение бинарного изображения на исходное с целью формирования выходного полутонового изображения с выделенными областями дубликатов.

Пусть максимальное значение кодов, которое может сформировать используемый локальный шаблон, равно N_{max} , тогда в каждом конкретном случае гистограмма H представляется в виде вектора, имеющего длину N_{max} . Анализ гистограммы заключается в поиске таких значений хэш-функции, для которых значение в гистограмме превышает единицу, то есть таких значений $l \in [0; N_{max}-1]$, что $H(l) > 1$. Далее принимается решение о принадлежности пикселей к числу дубликатов:

$$\forall i, j : M(i, j) = l \Rightarrow I(i, j) - \text{дубликат.} \quad (20)$$

3. Экспериментальные исследования

Для проведения экспериментальных исследований авторами используется стандартный ПК (Intel Core i5-4460 3.2 ГГц, 16 Гб ОЗУ). В качестве объектов исследования были выбраны 10 полутоновых цифровых изображений размером 512×512 . Авторами была разработана процедура генерации изображений, содержащих дубликаты, подвергавшиеся различным типам искажений (линейное контрастирование, импульсный шум). На рис. 7 показаны примеры изображений, содержащих искаженные дубликаты.

Сравнение проводилось между признаками на основе ЛБШ, ЦС-ЛБШ, УЛБШ, МБШ, ЛТШ, УЛТШ и ЛДШ в рамках алгоритма обнаружения дубликатов. Во всех экспериментах обработка осуществлялась окном 7×7 .

Для оценки качества разработанного алгоритма мы использовали величину $F1_Score$, которая связы-

вает точность обнаружения и ошибки первого и второго рода:

$$F1 = 2 \cdot tp / (2 \cdot tp + fp + fn), \quad (21)$$

где tp – количество верно обнаруженных дубликатов, fp – количество ложно обнаруженных дубликатов, fn – количество пропущенных дубликатов.



Рис. 7. Примеры генерируемых изображений: а) исходное изображение; б-в) изображения, содержащие дубликаты, искаженные при помощи: б) линейного контрастирования; в) аддитивного импульсного шума

Результаты работы исследуемого алгоритма на изображениях, содержащих в себе дубликаты с внесенными искажениями типа «линейное контрастирование», представлены в табл. 1 и на рис. 8.

Табл. 1. Исследование алгоритмов обнаружения дубликатов с искажениями типа линейное контрастирование

		а	0,9	0,7	0,5	0,3	0,1
Алгоритм							
	ЛБШ	32×32	0,42	0,15	0,10	0,08	0,10
	64×64		0,57	0,30	0,20	0,16	0,09
ЦС-ЛБШ	32×32		0,34	0,34	0,31	0,01	0,01
	64×64		0,63	0,54	0,52	0,30	0,05
УЛБШ	32×32		0,15	0,09	0,07	0,07	0,07
	64×64		0,44	0,43	0,46	0,29	0,23
МБШ	32×32		0,39	0,38	0,35	0,18	0,06
	64×64		0,68	0,59	0,45	0,43	0,21
ЛТШ	32×32		0,55	0,44	0,40	0,06	0,35
	64×64		0,79	0,51	0,50	0,44	0,40
УЛТШ	32×32		0,07	0,04	0,03	0,02	0,03
	64×64		0,25	0,16	0,11	0,07	0,11
ЛДШ-2	32×32		0,39	0,36	0,34	0,24	0,12
	64×64		0,63	0,58	0,50	0,35	0,24
ЛДШ-3	32×32		0,28	0,25	0,19	0,06	0,05
	64×64		0,31	0,29	0,31	0,22	0,17

Очевидно, что качество обнаружения зависит от размера области-дубликата: значение показателя $F1$ тем больше, чем больше размер области-дубликата.

Как видно из диаграммы, представленной на рис. 8, при значениях мультипликативного параметра a , превышающих 0,5, наилучшие результаты показывают алгоритмы, использующие МБШ и ЛТШ. При небольших значениях мультипликативного параметра линейного контрастирования алгоритм, использующий ЛТШ, дает более высокое значение показателя $F1$, чем алгоритмы, использующие другие виды локальных шаблонов. Также на диаграмме показано, что ЦС-ЛБШ, УЛБШ и МБШ инвариантны к линейному контрастированию при значениях параметра линейного контрастирования a , не превышающих 0,5.

Примеры результатов обнаружения дубликатов размера 64×64 с искажением типа «линейное контрастирование» представлены на рис. 9.

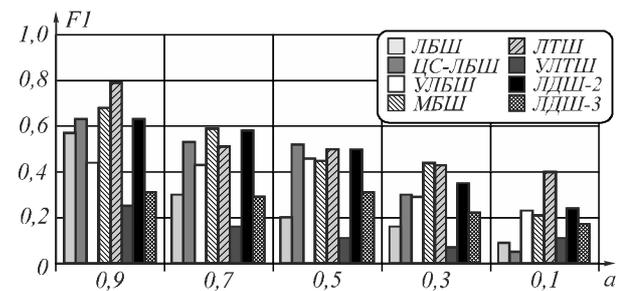


Рис. 8. Значения показателя $F1$ при использовании различных локальных шаблонов на изображениях с линейно-контрастированными дубликатами размера 64×64

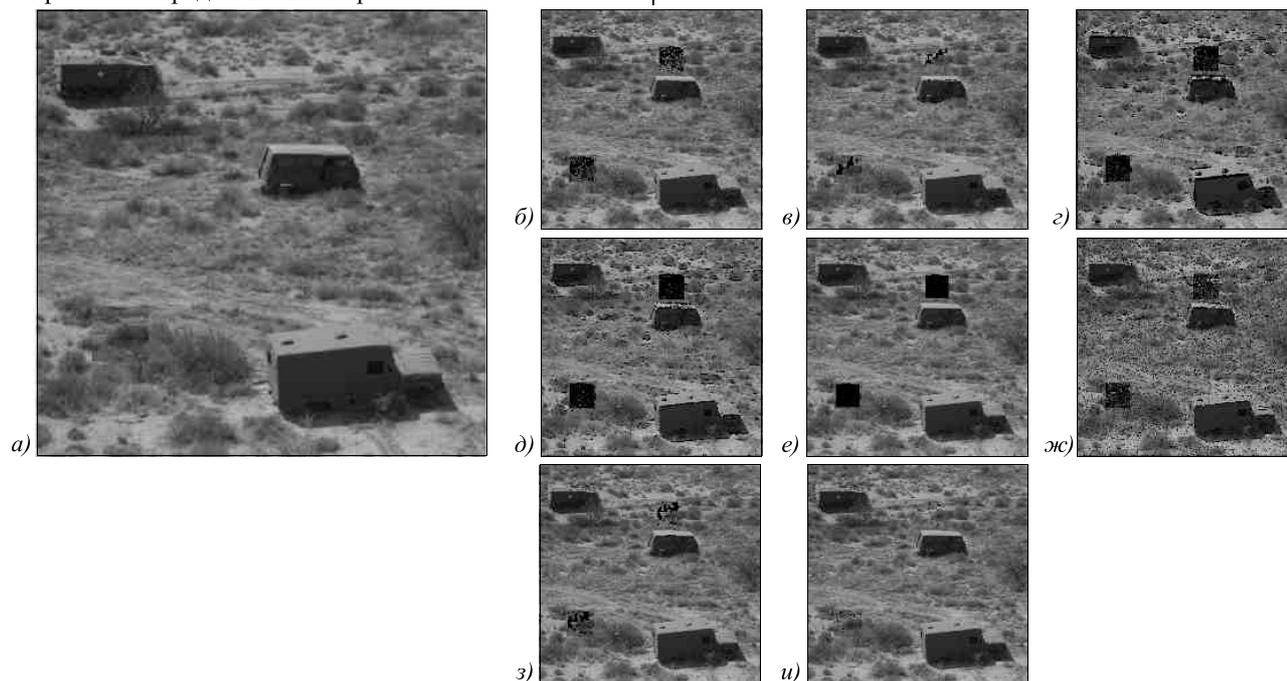


Рис. 9. Искаженное с помощью линейного контрастирования исходное изображение, $a = 0,8$ (а), результаты обнаружения дубликатов с искажением типа «линейное контрастирование» с использованием: ЛБШ (б), ЦС-ЛБШ (в), УЛБШ (г), МБШ (д), ЛТШ (е), УЛТШ (ж), ЛДШ-2 (з), ЛДШ-3 (и)

Результат применения исследуемых алгоритмов к изображениям, содержащим в себе дубликаты с внесенными искажениями типа «импульсный шум» с варьируемым значением параметра плотности шума $d \in [0,002; 0,1]$. Результаты работы алгоритма по всем используемым изображениям обобщены и приведены в табл. 2.

Анализируя диаграмму, представленную на рис. 10, можно сделать вывод, что почти все шаблоны позволяют получить достаточно высокие значения $F1$ при небольших значениях параметра плотности импульсного шума d . Однако значение показателя $F1$ при использовании ЛДШ второго порядка и ЛДШ третьего порядка выше, чем при использовании остальных локальных шаблонов. При значениях параметра d , превышающих 0,01, алгоритм с использованием МБШ дает более высокое значение показателя $F1$, чем алгоритм, использующий другие виды локальных шаблонов. Использование ЦС-ЛБШ дает худшие резуль-

таты в связи с его неустойчивым поведением при наличии шума.

Табл. 2. Исследование алгоритмов обнаружения дубликатов с искажениями типа импульсный шум

Алгоритм \ d		0,002	0,01	0,02	0,03	0,05
ЛБШ	32×32	0,55	0,49	0,27	0,13	0,04
	64×64	0,79	0,60	0,41	0,27	0,08
ЦС-ЛБШ	32×32	0,30	0,05	0,03	0,02	0,01
	64×64	0,56	0,24	0,07	0,01	0,01
УЛБШ	32×32	0,35	0,22	0,07	0,02	0,00
	64×64	0,66	0,35	0,24	0,16	0,04
МБШ	32×32	0,53	0,36	0,22	0,13	0,04
	64×64	0,77	0,61	0,47	0,29	0,15
ЛТШ	32×32	0,60	0,34	0,18	0,11	0,02
	64×64	0,72	0,54	0,30	0,20	0,05
УЛТШ	32×32	0,61	0,40	0,28	0,22	0,03
	64×64	0,79	0,60	0,39	0,21	0,07

ЛДШ-2	32×32	0,67	0,61	0,38	0,24	0,03
	64×64	0,82	0,59	0,35	0,23	0,09
ЛДШ-3	32×32	0,71	0,42	0,34	0,23	0,04
	64×64	0,79	0,64	0,38	0,23	0,07

Примеры результатов обнаружения дубликатов размера 64×64 с искажением типа «импульсный шум» представлены на рис. 11.

Выводы

В работе предложен обзор различных вариантов построения локальных шаблонов для решения задачи обнаружения искаженных дубликатов. Проведенные исследования позволили определить наилучшие в смысле выбранной метрики *F1* типы шаблонов, которые можно использовать для обнаружения дубликатов, искаженных при помощи линейного контрастирования и аддитивного шума. Так, для обнаружения дубликатов, подвергавшихся искажениям в виде линейного контрастирования, предлагается использовать ЛТШ в составе алгоритма обнаружения, а в случае искажения дубликатов при помощи аддитивного

шума лучше всех показали себя ЛДШ-2 и ЛДШ-3. В ходе исследований были также определены ограничения на параметры искажений, которые способен выявить алгоритм обнаружения дубликатов.

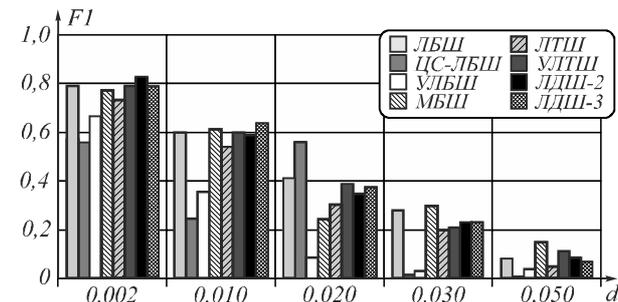


Рис. 10. Значения показателя *F1* при использовании различных локальных шаблонов на изображениях с дубликатами, искаженными импульсным шумом, размера 64×64

Благодарности

Работа выполнена при частичной финансовой поддержке гранта РФФИ № 16-37-00056.

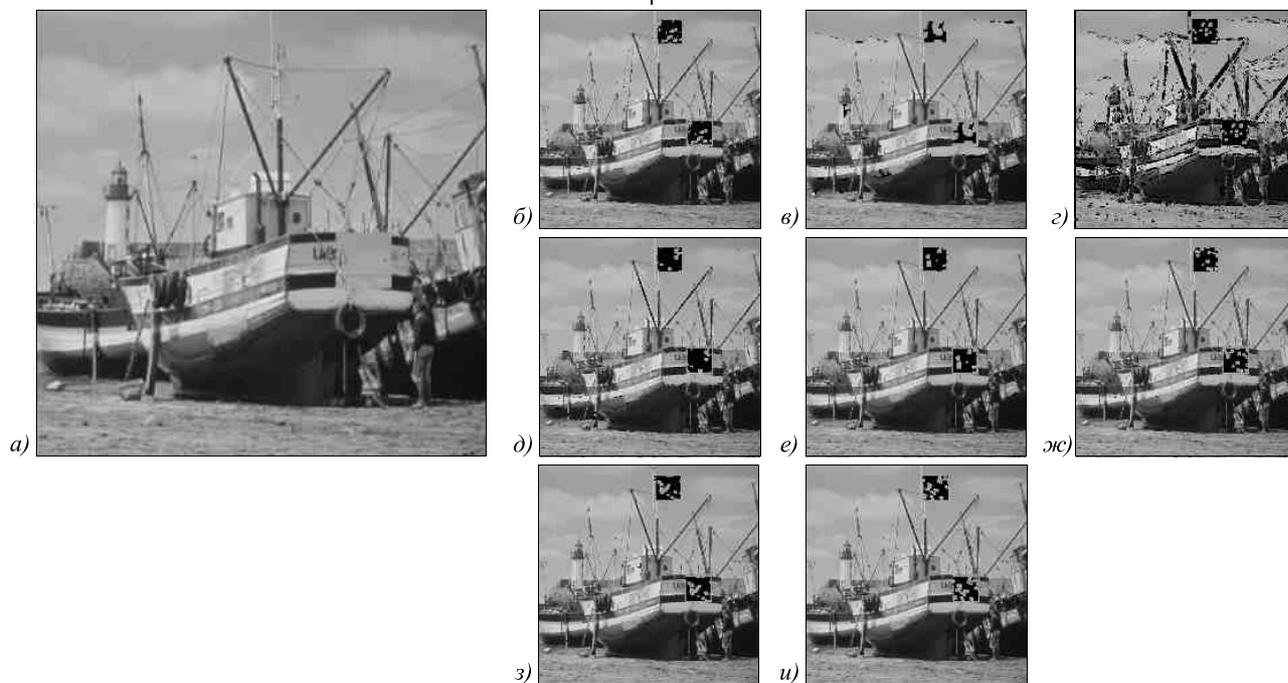


Рис. 11. Искаженное с помощью импульсного шума исходное изображение, $d = 0,005$ (а), результаты обнаружения дубликатов с искажением типа «импульсный шум» с использованием: ЛБШ (б), ЦС-ЛБШ (в), УЛБШ (г), МБШ (д), ЛТШ (е), УЛТШ (ж), ЛДШ-2 (з), ЛДШ-3 (и)

Литература

1. **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. – DOI: 10.1109/TIFS.2012.2218597.
2. **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 13.04.2016).
3. **Fridrich, J.** Detection of copy-move forgery in digital images [Electronical Resource] / J. Fridrich, D. Soukal, J. Lukáš. – 2006. – URL: <http://www.ws.binghamton.edu/fridrich/Research/copymove.pdf> (Request date 27.11.2016).
4. **Wang, L.** Texture classification using texture spectrum / L. Wang, D.-C. He // Pattern Recognition. – 1990. – Vol. 23(8). – P. 905-910. – DOI: 10.1016/0031-3203(90)90135-8.
5. **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. – DOI: 10.1109/TIP.2013.2268976.
6. **Heikkilä, M.** Description of interest regions with local binary patterns / M. Heikkilä, M. Pietikäinen, C. Schmid //

- Pattern Recognition. – 2009. – Vol. 42(3). – P. 425-436. – DOI: 10.1016/j.patcog.2008.08.014.
7. **Jin, H.** Face detection using improved LBP under bayesian framework / H. Jin, Q. Liu, H. Lu, X. Tong // Proceedings of the 3rd International Conference on Image and Graphics. – 2004. – P. 306-309. – DOI: 10.1109/ICIG.2004.62.
 8. **Hafiane, A.** Median binary pattern for textures classification / A. Hafiane, G. Seetharaman, B. Zavidovique // Proceedings of the 4th International Conference on Image Analysis and Recognition (ICIAR '07). – 2007. – P. 387-398. – DOI: 10.1007/978-3-540-74260-9_35.
 9. **Tan, X.** Enhanced local texture feature sets for face recognition under difficult lighting conditions / X. Tan, B. Triggs // IEEE Transactions on Image Processing. – 2010. – Vol. 19(6). – P. 1635-1650. – DOI: 10.1109/TIP.2010.2042645.
 10. **Nanni, L.** A local approach based on a Local Binary Patterns variant texture descriptor for classifying pain states / L. Nanni, S. Brahmam, A. Lumini // Expert Systems with Application. – 2010. – Vol. 37(12). – P. 7888-7894. – DOI: 10.1016/j.eswa.2010.04.048.
 11. **Zhang, B.** Local derivative pattern versus local binary pattern: face recognition with high-order local pattern descriptor / B. Zhang, Y. Gao, S. Zhao, J. Liu // IEEE Transactions on Image Processing. – 2010. – Vol. 19(2). – P. 533-544. – DOI: 10.1109/TIP.2009.2035882.
 12. **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.
 13. **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. – In book: Image Analysis and Recognition: 11th International Conference, ICIAR 2014, Vilamoura, Portugal, October 22-24, 2014, Proceedings, Part I / Ed. by Aurélio Campilho, Mohamed Kamel. – 2014. – P. 461-468. – DOI: 10.1007/978-3-319-11758-4_50.

Сведения об авторах

Евдокимова Надежда Ивановна, родилась в 1994 году. В 2016 году окончила Самарский национальный исследовательский университет имени академика С.П. Королёва (Самарский университет) с отличием по специальности «Информационная безопасность автоматизированных систем». В настоящее время является аспирантом Самарского университета. Основные сферы научных интересов: обработка изображений, обнаружение искажений на цифровых изображениях, защита информации. Имеет две публикации.
E-mail: nadezh.evdomkimoval@gmail.com.

Кузнецов Андрей Владимирович, родился в 1987 году. В 2010 году окончил 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 Самарского Университета и научным сотрудником в ИСОИ РАН– филиале ФНИЦ «Кристаллография и фотоника» РАН. Круг научных интересов включает обработку и анализ изображений, распознавание образов, обнаружение искажений изображений, геоинформатику. Имеет 37 публикаций, в том числе 18 научных статей и 1 монографию.
Страница в интернете: <http://nil97.ssau.ru/employee/detail.php?ID=35>. E-mail: kuznetsoff.andrey@gmail.com.

ГРНТИ: 28.23.15

Поступила в редакцию 12 декабря 2016 г. Окончательный вариант – 20 января 2017 г.

LOCAL PATTERNS IN THE COPY-MOVE DETECTION PROBLEM SOLUTION

N.I. Evdokimova¹, A.V. Kuznetsov^{1,2}

¹*Samara National Research University, Samara, Russia,*

²*Image Processing Systems Institute of RAS – Branch of the FSRC “Crystallography and Photonics” RAS, Samara, Russia*

Abstract

Embedding of duplicates is one of commonly used methods of image forgery. During this process, an image fragment is copied and pasted to another position in the same image. This is performed to conceal some important part of the image. A copy-move forgery detection algorithm aims to recognize duplicated areas in the image. This algorithm is based on calculating the characteristics in a sliding or overlapping window. In this paper, we compare the performance of copy-move detection algorithms that utilize a local binary pattern, a local ternary pattern, a local derivative pattern, and some extensions thereof. A distinctive feature of the used characteristics is their resistance to distortions inserted into the copy, such as linear contrast enhancement and impulse noise. This method also has low computational complexity.

Keywords: copy-move, forgery, local binary pattern, local ternary pattern, local derivative pattern.

Citation: Evdokimova NI, Kuznetsov AV. Local patterns in the copy-move detection problem solution. Computer Optics 2017; 41(1): 79-87. DOI: 10.18287/2412-6179-2017-41-1-79-87.

Acknowledgements: The work was partially funded by the RFBR grant No. 16-37-00056.

References

- [1] 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. DOI: 10.1109/TIFS.2012.2218597.
- [2] Popescu A, Farid H. Exposing digital forgeries by detecting duplicated image regions. Source: (<http://www.ists.dartmouth.edu/library/102.pdf>).
- [3] Fridrich J, Soukal D, Lukáš J. Detection of copy-move forgery in digital images. Source: (<http://www.ws.binghamton.edu/fridrich/Research/copymove.pdf>).
- [4] Wang L, He D-C. Texture classification using texture spectrum. *Pattern Recognition* 1990; 23(8): 905-910. DOI: 10.1016/0031-3203(90)90135-8.
- [5] 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. DOI: 10.1109/TIP.2013.2268976.
- [6] Heikkilä M, Pietikäinen M, Schmid C. Description of interest regions with local binary patterns. *Pattern Recognition* 2009; 42(3): 425-436. DOI: 10.1016/j.patcog.2008.08.014.
- [7] Jin H, Liu Q, Lu H, Tong X. Face detection using improved LBP under bayesian framework. *Proceedings of the 3rd International Conference on Image and Graphics* 2004; 306-309. DOI: 10.1109/ICIG.2004.62.
- [8] Hafiane A, Seetharaman G, Zavidovique B. Median binary pattern for textures classification. *ICIAR '07* 2007; 387-398. DOI: 10.1007/978-3-540-74260-9_35.
- [9] Tan X, Triggs B. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE Transactions on Image Processing* 2010; 19(6): 1635-1650. DOI: 10.1109/TIP.2010.2042645.
- [10] Nanni L, Brahnam S, Lumini A. A local approach based on a Local Binary Patterns variant texture descriptor for classifying pain states. *Expert Systems with Application* 2010; 37(12): 7888-7894. DOI: 10.1016/j.eswa.2010.04.048.
- [11] Zhang B, Gao Y, Zhao S, Liu J. Local derivative pattern versus local binary pattern: face recognition with high-order local pattern descriptor. *IEEE Transactions on Image Processing* 2010; 19(2): 533-544. DOI: 10.1109/TIP.2009.2035882.
- [12] Glumov NI, Kuznetsov AV, Myasnikov VV. The algorithm for copy-move detection on digital images [In Russian]. *Computer optics* 2013; 37(3): 360-367.
- [13] Kuznetsov A, Myasnikov V. A Fast Plain Copy-Move Detection Algorithm Based on Structural Pattern and 2D Rabin-Karp Rolling Hash. In book: Campilho A, Kamel M, eds. *Image Analysis and Recognition: 11th International Conference, ICIAR 2014, Vilamoura, Portugal, October 22-24, 2014, Proceedings, Part I* 2014: 461-468. DOI: 10.1007/978-3-319-11758-4_50.

Authors' information

Nadezhda Ivanovna Evdokimova (b. 1994) graduated with honors (2016) from Samara National Research University, Computer Science faculty. Programme – Information Security of Automated Systems. Nowadays she is postgraduate at Samara National Research University. Main research interests: image processing, digital image forgery detection, information security. She has two publications. E-mail: nadezh.evdokimova@gmail.com.

Andrey Vladimirovich Kuznetsov (b. 1987) graduated with honors (2010) from 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 Samara National Research University and a researcher at IPSI RAS– Branch of the FSRC “Crystallography and Photonics” 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. Web-page: <http://nil97.ssau.ru/employee/detail.php?ID=35>. E-mail: kuznetsoff.andrey@gmail.com.

Received December 12, 2016. The final version – January 20, 2017.