ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ ОБНАРУЖЕНИЯ ДУБЛИКАТОВ С ИСПОЛЬЗОВАНИЕМ ДЕРЕВЬЕВ ДВОИЧНОГО РАЗБИЕНИЯ ПРОСТРАНСТВА

А.В. Кузнецов, Е.В. Мясников

Самарский государственный аэрокосмический университет имени академика С.П. Королёва (национальный исследовательский университет) (СГАУ), Самара, Россия

Встраивание дубликатов является одним из самых часто применяемых методов сокрытия информации на цифровых изображениях. 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. Существующие подходы к поиску искажённых таким способом областей состоят из двух ключевых этапов: вычисление векторов признаков в рамках окна обработки с перекрытием и поиск близких векторов в евклидовом пространстве с применением лексикографической сортировки или kd-дерева. В данной работе мы предлагаем использовать на этапе поиска другой вид деревьев двоичного разбиения пространства (*binary space partitioning tree*) – vp-дерево. В работе представлено сравнение скорости поиска с его помощью и с помощью kd-дерева. Результаты демонстрируют преимущество предлагаемого подхода перед kd-деревом.

Ключевые слова: дубликат, искажение, kd-дерево, vp-дерево, дерево двоичного разбиения пространства.

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. Такой колоссальный объём данных не может не интересовать злоумышленников, целью которых является предоставление недостоверных данных конечному потребителю. 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 является актуальной задача разработки методов и алгоритмов обнаружения искусственных искажений цифровых изображений. Актуальность проблемы обуславливается также динамичным ростом числа программных продуктов, предназначенных для обработки цифровых изображений. Для использования такого программного обеспечения не требуется специальных навыков или обучения.

Среди всех существующих способов подделывания изображений наиболее часто применяемым 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. Широкое использование именно этой операции внесения искажений обуславливается простотой её выполнения.

В настоящее время существует большое количество работ, посвящённых разработке алгоритмов обнаружения искажённых и неискажённых дубликатов [1-5]: 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-фукнций в скользящем окне и дальнейший поиск одинаковых значений при помощи хэш-таблицы [4-5]. Предложенные решения показали высокую эффективность при низкой вычислительной сложности.

В данной работе будем считать операцию построения признакового описания локальных областей изображения решённой и сосредоточимся на исследовании процедуры поиска близких векторов признаков. Работа построена следующим образом. В первой части представлено описание общей схемы работы алгоритмов обнаружения дубликатов. Вторая часть посвящена описанию деревьев двоичного разбиения пространства. Третья часть содержит результаты экспериментальных исследований вычислительной сложности предложенного метода поиска близких векторов признаков на различных объёмах их выборок.

1. Схема работы алгоритмов обнаружения дубликатов

В настоящее время известно большое количество работ, посвящённых разработке алгоритмов обнаружения дубликатов [1-3]. Большинство из них придерживаются единой схемы, представленной на рис. 1.

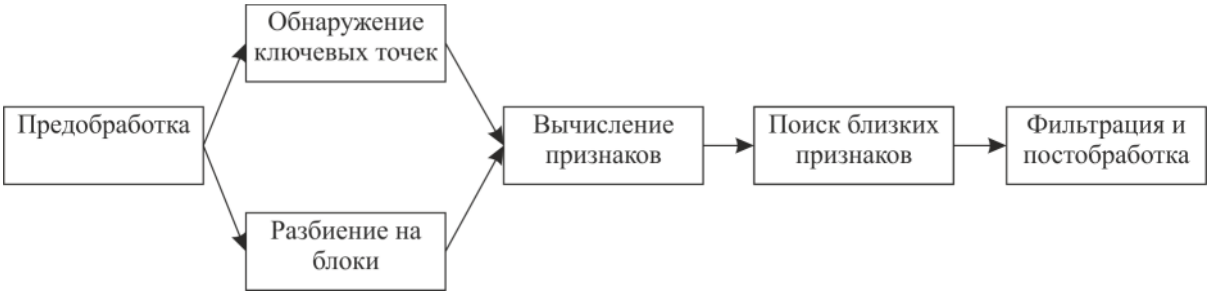

Рис.1. Общая схема работы алгоритмов обнаружения дубликатов

Первый этап алгоритма включает в себя предобработку анализируемого изображения. На этом шаге, как правило, производится фильтрация шума или объединение каналов многоканального изображения. Далее следует два альтернативных шага. Первый заключается в вычислении ключевых точек на изображении (например, в работе [3]), второй характеризует схему разбиения изображения на блоки для дальнейшего анализа окном обработки. Третий этап алгоритма посвящён вычислению векторов признаков либо на основе ключевых точек, либо в рамках окна обработки, определённого на предыдущем шаге. Следующий этап заключается в поиске наиболее близких векторов признаков, которые считаются потенциальными дубликатами. Данная процедура является наиболее ресурсоёмкой во всей схеме анализа изображения, поэтому от её выбора существенно зависит время работы всего алгоритма. Заключительным этапом алгоритма является фильтрация и постобработка. Данный шаг позволяет снизить количество ложно обнаруженных дуб-

ликатов и пропусков дубликатов, тем самым повысив значение показатель качества алгоритма анализа.

В большинстве работ по теме обнаружения дубликатов [1-3] большое внимание уделяется разработке и исследованию новых инвариантных к различным искажениям признаков, использование которых позволят повысить точность обнаружения. Задача улучшения скорости поиска близких векторов признаков в известных авторам работах не рассматривается. Очевидным решением является попарное сравнение всех векторов признаков, входящих в состав выборки. Такой подход является вычислительно неэффективным и обладает сложностью $O(N^2)$, что не позволяет применять его в задачах анализа большого объёма данных, например, при обработке данных дистанционного зондирования Земли. Для снижения вычислительной сложности в работах по обнаружению дубликатов авторы используют два основных метода: лексикографическую сортировку и kd-деревья.

Лексикографическая сортировка заключается в том, что все вектора признаков сначала помещаются в матрицу признаков построчно, а затем производится их сортировка по признакам последовательно. В итоге близкие вектора признаков будут расположены на соседних строках матрицы. Пример использования этого подхода продемонстрирован в работе [6].

В большей части работ [7-8] используется kd-дерево [9]. Оно применяется совместно с алгоритмом поиска k ближайших соседей, что позволяет сформировать для каждого вектора признаков список k ближайших к нему в Евклидовом пространстве векторов. В сравнении с лексикографической сортировкой этот подход приводит к лучшим результатам поиска и, как следствие, более точному обнаружению дубликатов.

В настоящей работе мы предлагаем использовать альтернативное дерево двоичного разбиения пространства - vp-дерево [10]. Предполагается, что использование vp-дерева позволит добиться лучших временных показателей по сравнению с kd-деревом. Такое предположение основывается на результатах проведенных ранее исследований, в которых vp-дерево показало лучшие результаты при решении таких задач, как поиск [10], а также снижение размерности [11] многомерных данных.

2. Деревья двоичного разбиения пространства

Kd-дерево [9] представляет собой сбалансированное дерево двоичного разбиения пространства, построение которого по множеству векторов осуществляется следующим образом. Среди всех осей координат пространства выбирается та, по которой будет осуществляться разбиение входного множества векторов. Выбор оси координат может выполняться циклически (сначала первая координата, затем вторая, и т.д.) или исходя из максимального разброса векторов в подмножестве вдоль выбираемой оси (в работе использовался последний вариант) После выбора оси координат среди множества векторов выбирается вектор с медианным значением по выбранной для разбиения координате. Выбранный вектор ставится в соответствие корню дерева, а все остальные вектора разделяются на два подмножества. В первое подмножество помещаются вектора со значением соответствующей координаты меньшим медианы, а во второе – со значением коор-

динаты большим медианы. Сформированные таким образом подмножества относятся к левому и правому поддеревьям корня соответственно, после чего процесс рекурсивно повторятся для левого и правого поддеревьев.

Vp-дерево [10] также представляет собой сбалансированное дерево двоичного разбиения пространства. Его построение по множеству векторов осуществляется следующим образом. Среди множества векторов выбирается один вектор (опорная точка, vantage point), который становится корнем дерева. Далее все остальные вектора разделяются на два подмножества таким образом, что в первом подмножестве оказываются вектора, удаленные от выбранной опорной точки на расстояние не превышающее некоторое значение R, а во втором подмножестве – вектора, удаленные от опорной точки на расстояние большее R. Выбор порогового значения R осуществляется так, чтобы количество элементов в левом и правом подмножествах отличалось не более, чем на единицу. Выделенные указанным образом подмножества относятся к левому и правому поддеревьям корня дерева соответственно, и описанный процесс для них рекурсивно повторяется.

Построенные по описанным выше алгоритмам деревья могут быть использованы для поиска ближайшего соседа с использованием метода ветвей и границ так, как описано в работах [9, 10]. Необходимо отметить, что рассмотренные структуры выполняют разбиение пространства существенно различными способами: гиперплоскостями, ортогональными осям координат в kd-деревьях и гиперсферами с центрами в опорных точках в vp-деревьях.

3. Экспериментальные исследования

Для проведения экспериментальных исследований авторами используется стандартный ПК (Intel Core i5-3470 3.2. ГГц, 4 Гб ОЗУ), реализация алгоритмов производилась на языке C++ в среде разработки Microsoft Visual Studio 2013. Объектом исследования являются наборы векторов признаков различного объёма от 5000 до 15000 элементов. Формирование векторов признаков проводилось на наборе изображений с размерами 512x512 следующим образом: в режиме обработки окном 16x16 с перекрытиями в 4 и более пикселей для каждого его положения строился вектор признаков на основе бинарных градиентных контуров [12].

В ходе эксперимента для каждого вектора признаков осуществляется поиск ближайшего вектора среди остальных элементов выборки. По окончании работы алгоритма формируется список пар индексов ближайших векторов и вычисляется время работы алгоритма поиска в секундах. Сравнение времени поиска проводится для трёх решений: поэлементного сравнения всех векторов признаков (brute force), наиболее часто используемого в задаче обнаружения дубликатов kd-дерева (kd-tree) и предложенного в данной работе дерева двоичного разбиения пространства (vp-tree). Результаты проведённого эксперимента, представленные в Таблице 1, показывают, что предложенный подход производит поиск на 1-2 секунды быстрее kd-дерева и значительно быстрее прямого попарного сравнения при увеличении размера выборки.

Табл.1. Сравнение времени поиска (sec.) ближайшего вектора признаков для размеров выборок от 5000 до 15000

	5000	**7500**	**10000**	**12500**	**15000**
Brute force	9,6	22,7	39,0	62,0	90,0
Kd-tree	5,7	13,4	24,5	36,3	49,0
vp-tree	4,6	11,0	22,1	34,6	47,2

Выводы

В рамках данного исследования был предложен альтернативный подход к поиску близких векторов признаков для решения задачи обнаружения дубликатов на цифровых изображениях с использованием vp-деревьев. Предложенное решение показало лучшее время поиска близких векторов в сравнении с наиболее часто используемым решением на основе kd-дерева. В дальнейшем планируется провести исследование эффективности применения vp-деревьев, построенных с использованием различных алгоритмов выбора опорных точек, для решения задачи поиска дубликатов.

Благодарности

Работа по исследованию методов поиска близких векторов признаков с использованием деревьев двоичного разбиения пространства выполнена в рамках грантов РФФИ 16-37-00056 мол_а и 16-37-00202 мол_а.

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.
2. Popescu, A. Exposing digital forgeries by detecting duplicated image regions / A. Popescu, H. Farid [Электронный ресурс]. – 2004. – URL: http://www.ists.dartmouth.edu/library/102.
3. Fridrich J. Detection of copy-move forgery in digital images / J. Fridrich, D. Soukal, J. Lukas. // Proceedings of Digital Forensic Research Workshop, August 2003.
4. 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.
5. Kuznetsov, A. A Fast Plain Copy-Move Detection Algorithm Based on Structural Pattern and 2D Rabin-Karp Rolling Hash / A. Kuznetsov, V. Myasnikov // Lecture Notes in Computer Science. – 2014. – Vol. 8814(1), P. 461-468.
6. Bayram, S. Image Manipulation Detection with Binary Similarity Measures / S. Bayram, I. Avcıbas¸, B. Sankur, N. Memon // European Signal Processing Conference. – 2005.
7. 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. – P. 272–276.
8. Ju, S. An Authentication Method for Copy Areas of Images / S. Ju, J. Zhou, K. He // International Conference on Image and Graphics. – 2007. – P. 303–306.
9. Bentley, J. L. Multidimensional binary search trees used for associative searching / J. L. Bentley // Communications of the ACM, 18 (9). – 1975. – 509.
10. Yianilos Data structures and algorithms for nearest neighbor search in general metric spaces / Yianilos // Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms. - 1993. - pp. 311–321.
11. Myasnikov, E.V. Evaluation of Space Partitioning Data Structures for Nonlinear Mapping / E.V. Myasnikov // Computer Science Research Notes. WSCG Full Papers Proceedings, 8-12 June 2015 - P. 109-118.
12. 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. – (в печати).