УДК 004.932
DOI: 10.15827/0236-235X.135.420-432
Дата подачи статьи: 08.04.21
2021. Т. 34. № 3. С. 420–432

Способы ускорения подготовки и встраивания цифрового водяного знака с использованием мобильных устройств на основе преобразования Арнольда и вейвлет-преобразования

А.Г. Зотин [1], *к.т.н., доцент, zotin@sibsau.ru*
А.В. Проскурин [1], *к.т.н., доцент, proskurin.av.wof@gmail.com*

[1] *Сибирский государственный университет науки и технологий им. академика М.Ф. Решетнева, г. Красноярск, 660037, Россия*

В последние годы технология цифрового маркирования фото- и видеоматериалов приобретает все большее значение из-за взрывного роста объемов информации, передаваемой через незащищенные каналы связи. Встроенные с помощью этой технологии цифровые водяные знаки позволяют сократить объемы передаваемой информации, а также могут применяться для защиты изображений (носителей) от незаконного использования. Для более эффективной реализации последней задачи проводятся различные исследования с целью повышения устойчивости, незаметности и защищенности цифровых водяных знаков. В большинстве случаев это приводит к увеличению требуемой вычислительной мощности, что затрудняет применение цифрового маркирования в мобильных устройствах.

В данной работе предложены несколько способов снижения вычислительных затрат и уменьшения времени вычислений методов цифрового маркирования, основанных на преобразовании Арнольда и вейлет-преобразовании. Первый способ заключается в линейной интерпретации цифровых водяных знаков и изображения-носителя, что позволяет избежать использования двойных циклов. Второй способ состоит в применении таблиц преобразований для замены непосредственных вычислений. Одна из таких таблиц позволяет выполнять преобразование Арнольда за определенное время вне зависимости от количества итераций. Для определения количества итераций для каждого блока используются хэш-код секретного ключа и специально сформированные для этого таблицы. Третий способ сокращения времени встраивания цифровых водяных знаков состоит в многопоточном выполнении, реализованном с помощью технологии OpenMP. В совокупности с применением линейной интерпретации это дает ускорение в 1,90, 2,56 и 3,01 раза для двух, трех и четырех потоков соответственно.

Ключевые слова*: цифровые водяные знаки, ЦВЗ, преобразование Арнольда, таблицы преобразования, OpenMP.*

В связи с активным развитием мобильных и сетевых технологий в последние два десятилетия все большие объемы мультимедийной информации передаются через незащищенные каналы связи. При этом изображения и видео можно свободно копировать, редактировать и распространять, что затрудняет доказательство их авторства. Один из способов решения этой проблемы заключается в использовании *цифровых водяных знаков* (ЦВЗ). При нанесении ЦВЗ секретная информация, которая обычно представлена в виде небольшого изображения, скрывается внутри основного изображения, называемого носителем, с минимальными визуальными искажениями последнего. При этом ЦВЗ может быть извлечен обратно в исходном виде, что позволяет использовать его в качестве доказательства авторства. Таким образом, ЦВЗ и алгоритм его встраивания должны обладать следующими свойствами [1, 2]:

– незаметность – встраивание ЦВЗ не должно приводить к очевидным визуальным искажениям носителя, а сама скрытая информация быть заметной человеку;

– устойчивость – распространенные атаки на носитель, такие как сжатие алгоритмом JPEG, фильтрация, обрезка или зеркальное отображение, не должны приводить к существенным искажениям ЦВЗ и затруднять его извлечение;

– вместимость – в носитель необходимо встроить как можно больше скрытой информации, продублировав ее для повышения вероятности успешного извлечения или добавив дополнительную информацию об авторе;

– низкая вычислительная стоимость – мобильное устройство должно встраивать ЦВЗ в изображение высокого разрешения за приемлемое время.

Существуют два основных подхода к встраиванию ЦВЗ – встраивание информации в про-

странственную или частотную область носителя. Пространственные методы основаны на прямом изменении параметров пикселов в выбранном регионе носителя. В качестве параметров могут выступать яркость или интенсивность цветовых каналов RGB. Наиболее известные пространственные методы – наименьшего значащего бита и его модификации [3, 4], а также средних значащих битов и его модификации [5, 6]. Данные методы просты в реализации и позволяют встроить большой объем информации. Однако встроенные ЦВЗ легко обнаруживаются с помощью компьютерного анализа или визуально. Кроме того, эти методы встраивания не способны эффективно противостоять большинству типов атак. Частотные методы обладают более высокой устойчивостью, поскольку ЦВЗ внедряется в частотные коэффициенты носителя (изображения). При этом определение частотных коэффициентов может происходить с использованием различных преобразований: дискретного преобразования Фурье [7], дискретного косинусного преобразования [8], *дискретного вейвлет-преобразования* (ДВП) и его модификаций [9, 10], сингулярного разложения [11]. Встраивание ЦВЗ в область средних частот позволяет одновременно повысить незаметность и устойчивость знака. Однако объем встраиваемой информации в таком случае существенно ниже, а необходимые вычислительные затраты значительно выше, чем при использовании пространственных методов. Несмотря на это, в последние годы широкое распространение получают именно частотные методы, так как они устойчивы ко многим видам атак [12].

С целью повышения устойчивости и незаметности многие методы используют предварительную обработку ЦВЗ алгоритмами скремблирования. Данные алгоритмы основаны на итерационном изменении положения пикселов изображения посредством матричного преобразования, что позволяет достичь хаотического визуального эффекта. Это дает два положительных эффекта. С одной стороны, скремблирование позволяет равномерно распределять биты ЦВЗ по всему изображению, что повышает устойчивость к таким атакам, как обрезка, шум, сжатие и фильтрация, а также затрудняет обнаружение ЦВЗ с помощью компьютерного анализа. С другой – скремблирование может повысить безопасность передачи секретной информации с помощью ЦВЗ, задавая количество итераций как ключ шифрования. В таком случае только владельцы знают секретный ключ для восстановления ЦВЗ и исходного носителя. В связи с этим достаточно активно разрабатываются новые схемы использования алгоритмов скремблирования при встраивании ЦВЗ. Среди существующих алгоритмов широкое распространение получили преобразование Арнольда [11, 13], отображение пекаря (baker's map) [14], логистическая хаотическая карта (logistic chaotic map) и преобразование магическим квадратом (magic square transform) [15]. Данные алгоритмы могут быть расширены для использования на разных цветовых каналах и в частотной области изображения. Негативной стороной скремблирования является высокая вычислительная сложность, вызванная необходимостью итеративной обработки всех элементов двумерной матрицы.

Некоторые из разработанных методов предполагают дополнительные шаги при подготовке ЦВЗ, направленные на повышение надежности и безопасности передачи данных. Например, в статье [16] текстовая информация кодируется с помощью штрихкода Code 128 для повышения вероятности считывания информации даже при сильных повреждениях ЦВЗ. К полученному изображению штрихкода применяется преобразование Арнольда. После этого биты ЦВЗ встраиваются в области, полученные в ходе двухуровневого ДВП цветового канала Cb.

Данный метод позволяет корректно считывать информацию даже при 30-процентном повреждении водяного знака, но требует затрат на преобразования (включая конвертации из одной цветовой модели в другую).

Также в последние годы во многих исследованиях встраивание ЦВЗ осуществляется с использованием комбинаций частотных преобразований. Например, в работе [9] предложен метод, комбинирующий гомоморфное преобразование, *дискретное избыточное вейвлет-преобразование* (ДИВП), преобразование Арнольда и сингулярное разложение. ДИВП применяется к носителю для получения области LL, которая разделяется на компоненты освещения и отражения посредством гомоморфного преобразования. С целью повышения безопасности в такой схеме используется преобразование Арнольда для скремблирования водяного знака, встраиваемого с помощью сингулярных значений компоненты отражения. Данный метод демонстрирует превосходную незаметность и устойчивость ЦВЗ, однако требует огромных вычислительных затрат.

Таким образом, методы, основанные на частотном преобразовании и скремблировании, устойчивы ко многим видам атак, однако требуют больших вычислительных затрат. Проводимые исследования, большинство из которых направлено на повышение незаметности и устойчивости ЦВЗ, только увеличивают эти затраты, что затрудняет использование технологии водяных знаков в мобильных устройствах. Для решения этих проблем в данной работе предложены улучшения, направленные на снижение общих вычислительных затрат, а также шифрование данных при подготовке и встраивании ЦВЗ.

Встраивание и извлечение информации на основе ДВП

Авторами работы за основу был взят метод встраивания ЦВЗ, описанный в статье [16] (далее – базовый). Исходными данными метода являются изображение или кадр для встраивания (носитель), текстовая информация для встраивания и секретный ключ. Общая схема встраивания текстовых данных в изображения приведена на рисунке 1.

Условно в схеме встраивания информации можно выделить три ключевых этапа:

– подготовка информации и формирование ЦВЗ;

– определение схемы встраивания;

– встраивание ЦВЗ в носитель.

В процессе подготовки ЦВЗ выполняется преобразование исходной текстовой информации с учетом секретного ключа. Для повышения устойчивости ЦВЗ к различным видам атак [12] используется алгоритм преобразования данных в штрихкод (Code 128). Полученное отображение штрихкода имеет ширину модуля в 1 пиксел и высоту 16 пикселов. Для осуществления скремблирования штрихкод делится на сегменты размером 32×16 пикселов, после чего формируются квадратные блоки размером 32×32. К данным блокам применяется разное количество преобразований Арнольда, количество итераций определяется на основе секретного ключа. После преобразования блоки объединяются, формируя код ЦВЗ для встраивания. Если емкость носителя больше встраиваемого ЦВЗ, происходит циклическая запись.

Определение схемы встраивания подразумевает формирование наборов частотных ко-

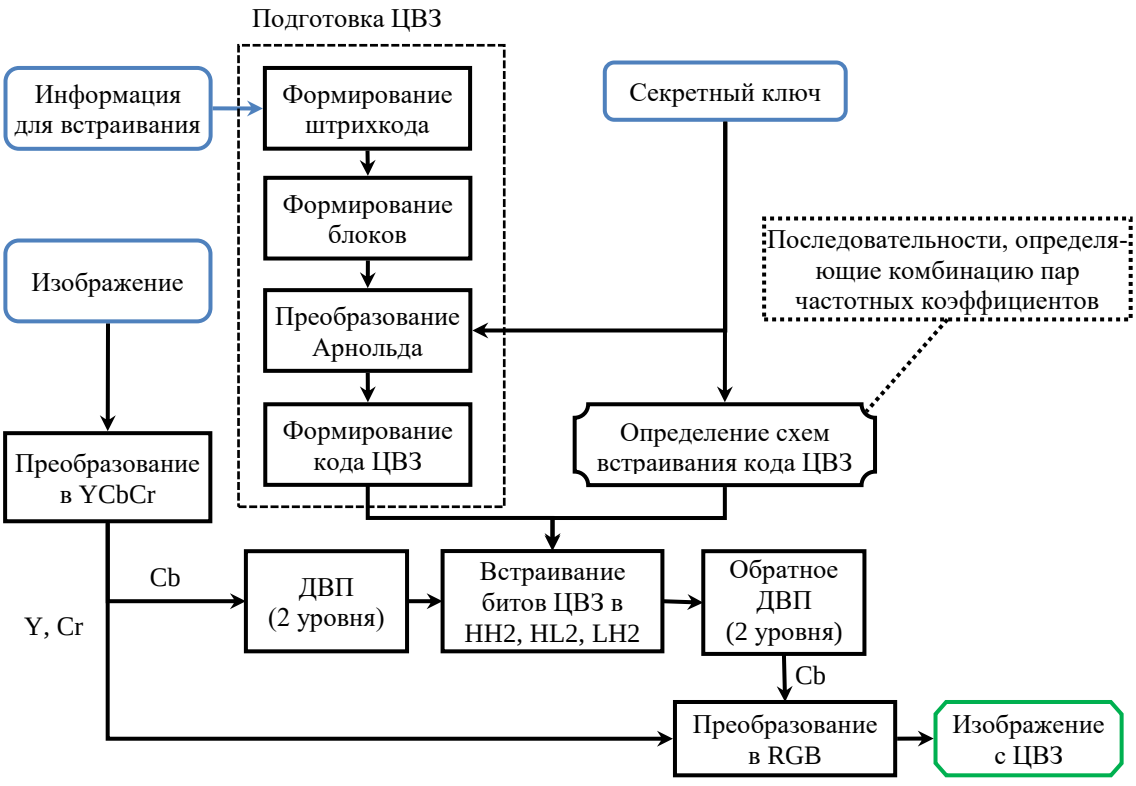


Рис. 1. Схема процесса встраивания информации в изображение

Fig. 1. A flow-chart of the watermark embedding method

эффициентов на основе секретного ключа. Для каждой частотной области HH2, HL2 и LH2 формируется свой уникальный список коэффициентов, которые будут использованы для встраивания битов слепка ЦВЗ. Весь процесс встраивания битов ЦВЗ происходит по следующей схеме:

– преобразование носителя из цветовой модели RGB в YCbCr;

– применение двухуровного дискретного преобразования Хаара к цветовому каналу Cb;

– встраивание ЦВЗ с помощью модифицированного алгоритма Коха–Жао в HH2, HL2 и LH2 области;

– применение обратного двухуровневого дискретного преобразования Хаара;

– преобразование носителя из цветовой модели YCbCr в RGB.

Схема извлечения информационного ЦВЗ из носителя выполняется по схеме, представленной на рисунке 2.

Предлагаемые модификации

Работа рассмотренных схем во многом зависит от секретного ключа, с помощью которого определяются схемы встраивания/извлечения и количество итераций преобразования Арнольда. Ускорения подготовки ЦВЗ, а также процессов встраивания/извлечения его битов можно достичь применением различных механизмов.

Для упрощения распределения данных в памяти и сокращения количества двумерных циклов решено использовать линейную интерпретацию данных. При такой организации данных появляется возможность применения таблиц преобразования и более эффективного использования механизмов распараллеливания.

При программной реализации использование непосредственного значения ключа нецелесообразно из-за наличия сложных правил с множеством условий. Использование условий, которые в программном коде реализуются при помощи механизмов ветвления, негативно сказывается на быстродействии алгоритмов. В связи с этим решено использовать таблицы преобразования на ограниченном наборе возможных значений. Набор допустимых значений в таблицах преобразований будет ограничиваться значением хэш-кода секретного ключа, полученного применением алгоритмов MD5, SHA-256, SHA-384 или SHA-512 к секретному ключу. Примеры последовательностей хэш-кодов для ключа Test@Key# Watermark приведены в таблице 1.

Символ кода принимает значение в виде цифр 0–9 и латинских букв A–F, которые в совокупности можно интерпретировать как значения 0–15. В этом случае скремблирование

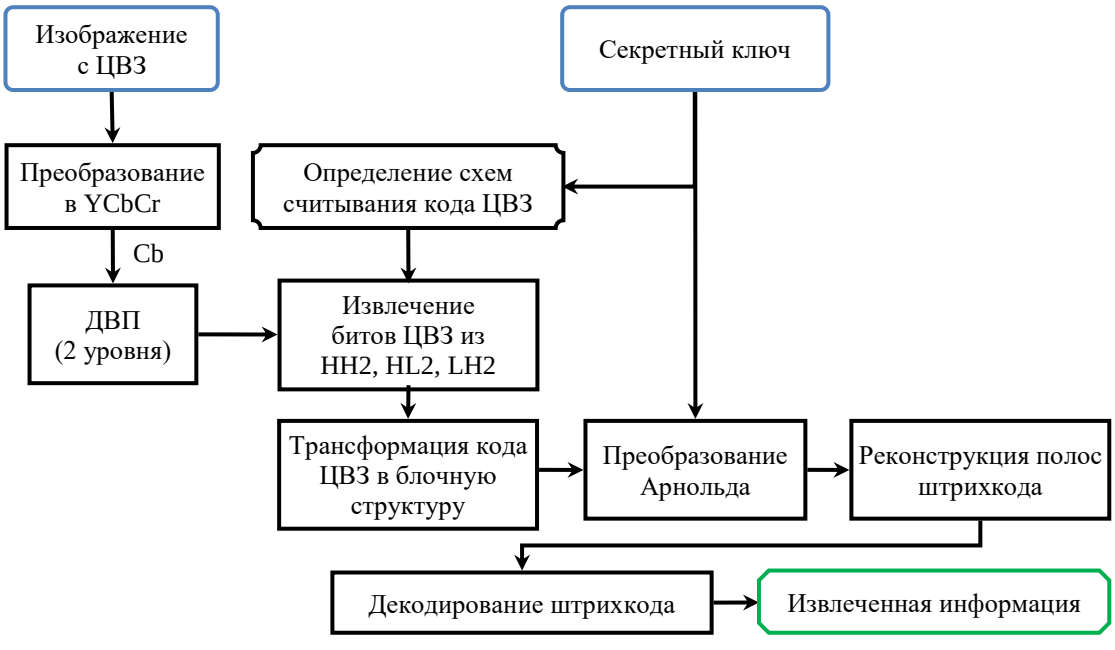


Рис. 2. Схема процесса извлечения информации из изображения

Fig. 2. A flow-chart of the watermark extraction method

Таблица 1

Примеры различных хэш-кодов для ключа

Table 1

The examples of different hash codes for a key

Хэш-функция	Хэш-код
MD5	B17740EE08AED1A996328C3081A8537C
SHA-256	8140778612769DBAB2A6A874B535C0AD76206F41C18CAEC54A2BD8492B80B122
SHA-384	F4C9ABEDB1DD8940E2751E802F82FE5CE376D919E72218F684571F97E5F47F3E6 2932EF8BA81D96FCB7D1693D10450DD
SHA-512	446AB01A554D7B25143CB1AF94CD8084F5039EC28072FA406070410DC179F7603 E5AEA09B7B74E1B75A4F31F3962774961A2B5DB5D7E16B3BA523EE39A6E7C04

можно выполнить с помощью таблиц преобразования, что позволит добиться константного времени выполнения преобразования Арнольда. При этом для улучшения кодирования информации к каждому блоку будет применяться разное количество итераций, которое определяется на основе MD5 кода. Также было решено использовать таблицы преобразования для быстрой реализации ДВП и определения коэффициентов, использующихся при встраивании каждого бита информации в области вейвлет-преобразования.

Подготовка ЦВЗ и преобразование Арнольда

Подготовка ЦВЗ заключается в преобразовании встраиваемой информации в такую последовательность битов, которая позволила бы повысить надежность и безопасность передачи данных. Используемая в базовом методе схема формирования блоков ЦВЗ позволяет сохранить в каждом блоке лишь часть внедряемой информации. В случае искажения носителя восстановление данных в большей степени будет зависеть от цикличности повторения самого слепка ЦВЗ. Предлагаемая модификация, заключающаяся в линейной интерпретации всего входного набора данных, предполагает занесение в каждый блок частичной информации обо всем штрихкоде. Если ширина ЦВЗ меньше 1 024 пикселов, то один блок размером 32×32 пиксела включает сразу несколько фрагментов с цикличным повторением. Таким образом, можно повысить вероятность корректного восстановления штрихкода. Схема линейного считывания ЦВЗ представлена на рисунке 3.

На следующем шаге для лучшего сокрытия данных используется преобразование Арнольда. Это преобразование применяется только к квадратным изображениям, однако его можно применить и к полученным одномерным представлениям. Для блока размерностью $N \times N$ преобразование Арнольда изменяет координаты (X, Y) элемента в новые координаты (X_{new}, Y_{new}) согласно выражению

$$\begin{bmatrix} X_{new} \\ Y_{new} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} \bmod N. \quad (1)$$

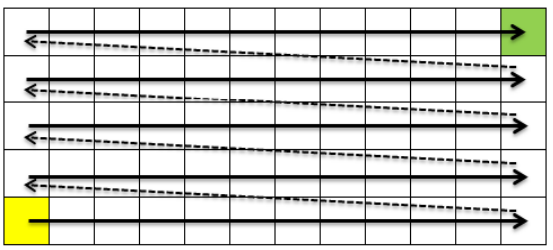


Рис. 3. Схема линейного считывания ЦВЗ

Fig. 3. Scheme of a watermark linear reading

Ключевой особенностью преобразования Арнольда является то, что после определенного количества итераций получается его оригинальное значение. В таблице 2 отражено количество итераций для типовых размеров блока.

Таблица 2

Количество итераций для блоков разного размера, после которых будет получен исходный блок

Table 2

The number of iterations for the blocks of different sizes, to obtain the original block

Размер блока	Количество итераций	Размер блока	Количество итераций
8×8	6	72×72	14
12×12	12	80×80	60
16×16	12	88×88	30
24×24	12	96×96	24
32×32	24	104×104	42
40×40	30	112×112	24
48×48	12	120×120	60
56×56	24	128×128	96
64×64	48	256×256	192

Пример трансформации блока размером 32×32 пиксела приведен на рисунке 4.

Непосредственное применение преобразования Арнольда приводит к высоким вычислительным затратам из-за итерирования по двумерной матрице. В связи с этим предлагается использовать таблицы преобразований. Для их формирования необходимо переписать выражение (1) в следующей форме:

$$X_{new} = (X + Y) \bmod N,$$
$$Y_{new} = (X + 2Y) \bmod N. \quad (2)$$

При формировании таблиц преобразования необходимо учитывать одномерную интерпретацию блока. С учетом уравнений (2) формирование таблицы *A*1*D* для одной итерации скремблирования будет происходить следующим образом: $A1D[YN + X] = Y_{new}N + X_{new}$.

Чтобы не повторять применение таблицы преобразования *A*1*D* многократно, дополнительно формируется таблица преобразований *A*2*D*, позволяющая добиться постоянного времени выполнения преобразования Арнольда вне зависимости от количества итераций. Первый уровень в таблице *A*2*D* означает номер итерации, второй – непосредственно параметры трансформации (аналогично *A*1*D*). Значения на первом уровне (*A*2*D*[1]) равны значениям из таблицы *A*1*D*. Расчет последующих уровней таблицы преобразования осуществляется согласно выражению $A2D[i][p] = A2D[i-1][A2D[1][p]]$, где i – номер текущей итерации преобразования; p – позиция параметра трансформации в таблице преобразования.

Каждый блок ЦВЗ преобразуется с помощью своего собственного количества итераций. Используемая в базовом методе схема определения количества итераций на основе непосредственных значений символов секретного ключа усложняла работу алгоритма. Если ключ небольшой, значения количества итераций регулярно повторяются. В связи с этим предложено сформировать таблицу преобразования AAlter, которая будет определять количество итераций преобразования, применяемых для блока. При ее генерации используются параметры хэш-кода, полученного для секретного ключа. Заполнение таблицы AAlter осуществляется в зависимости от допустимого набора итераций. Например, количество итераций может определяться с помощью базиса (первое число кода) и дополнительного смещения (остаток от суммы 1–3 последующих чисел). Применение MD5-кода даст возможность задать количество итераций для 16, 10 и 8 блоков размером 32×32 пиксела соответственно. При необходимости встраивать большее количество информации можно либо использовать иную хэш-функцию, либо задействовать цикличность.

Пример преобразования текстовой информации с помощью базового метода и с учетом предложенной модификации представлен на рисунке 5.

Определение схем встраивания и извлечения данных

По сравнению с базовым методом алгоритм встраивания ЦВЗ не претерпел существенных изменений. Для определения схемы встраивания битов авторами применяются линейная интерпретация частотной области, а также таблица преобразования WMX, построенная с помощью хэш-кодов секретного ключа и идентификатора частотной области.

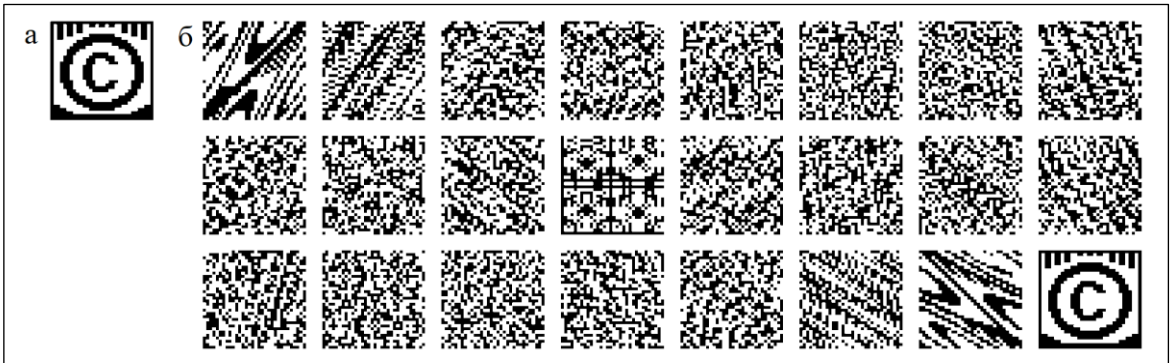


Рис. 4. Пример преобразования Арнольда для блока размером 32×32:
а) исходный блок, б) слева направо и сверху вниз – 24 итерации преобразования

Fig. 4. The example of Arnold transform for 32×32 block:
a) the original block, б) from left to right and from top to bottom – 24 transform iterations

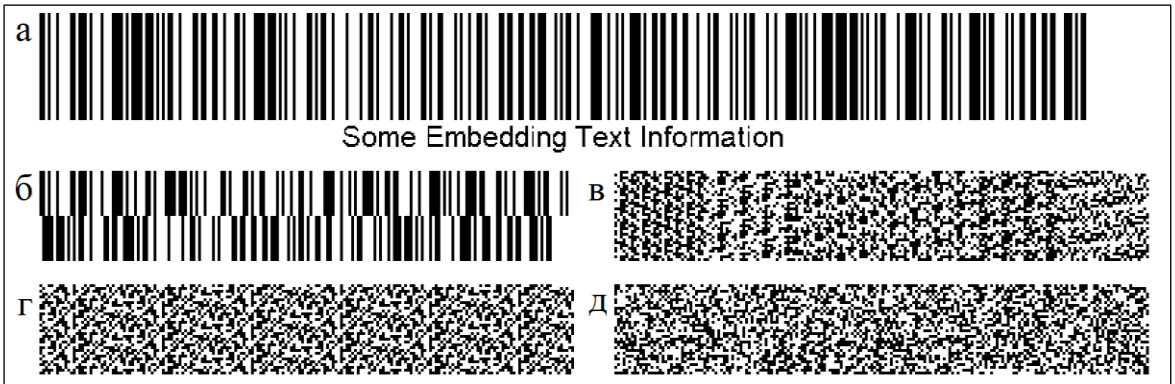


Рис. 5. Пример представления текстовой информации: а) исходный текст и его штрихкод, б) блочное представление штрихкода (базовый метод), в) блочное представление штрихкода после преобразования Арнольда (15 итераций), г) линейная интерпретация штрихкода, д) линейная интерпретация после преобразования Арнольда (используется секретный ключ)

Fig. 5. The example of text information presentation: a) the original text and its barcode, б) shrunk barcode (basic method), в) shrunk barcode after the Arnold transform (15 iterations), г) barcode linear interpretation (proposed method), д) linear interpretation after the Arnold transform (use of a secret key)

При вейвлет-преобразовании второго уровня для встраивания будут использованы два частотных коэффициента из четырех доступных. При этом в случае четырех коэффициентов {*P*1, *P*2, *P*3, *P*4} возможны 12 комбинаций, которые можно представить множеством ({*P*1, *P*2}, {*P*1, *P*3}, {*P*1, *P*4}, {*P*2, *P*1}, {*P*2, *P*3}, {*P*2, *P*4}, {*P*3, *P*1}, {*P*3, *P*2}, {*P*3, *P*4}, {*P*4, *P*1}, {*P*4, *P*2}, {*P*4, *P*3}). Базовая часть таблицы WMX заполняется на основе этих 12 комбинаций. В зависимости от того, в какую область вейвлет-коэффициентов (*HH*2, *HL*2 или *LH*2) будет встраиваться ЦВЗ, порядок комбинаций может быть разнообразным. Для этого осуществляется циклический сдвиг на основе хэш-кода, полученного для секретного ключа и выбранной частотной области. Для определения параметров сдвига базовой части используется остаток от деления первого символа на 12. Тип сдвига определяется на основе остатка от деления второго символа на 3 (0 – все элементы, 1 – нечетные элементы, 2 – четные элементы).

Дополнительно таблица преобразований WMX расширяется до 16 пар (некоторые пары будут повторяться) для простого использования значений символов хэш-кода. Дублирующие пары определяются с помощью первых четырех символов MD5 хэш-кода ключа. Пример формирования таблицы WMX для частотной области LH2 показан на рисунке 6. На основе полученных для каждой частотной области таблиц WMX с применением SHA-256, SHA-384 или SHA-512 хэш-кодов определяются последовательности используемых коэффициентов.

Применение параллельных вычислений

Для достижения большего ускорения при встраивании ЦВЗ возможно применение технологии распараллеливания обработки. Наиболее подходящим видом распараллеливания программного кода применительно к подготовке ЦВЗ, осуществления вейвлет-преобразования, а также непосредственного встраивания/извлечения ЦВЗ является распараллеливание, учитывающее параллелизм данных.

Такому виду распараллеливания соответствуют задачи, которые включают неоднократное выполнение одного и того же алгоритма с различными исходными данными. Вычисления могут производиться параллельно в случае разделения данных на фрагменты и обработки каждого фрагмента выделенным ядром. Для реализации параллельных алгоритмов широкое распространение получил стандарт OpenMP [17], применяемый для распараллеливания программ на языках C, C++ и Фортран. Распараллеливание в OpenMP выполняется явно путем написания в коде специальных директив, а также вызова вспомогательных функций.

С учетом измененного представления данных в виде линейной формы (одномерный масив) можно получить ускорение обработки по-

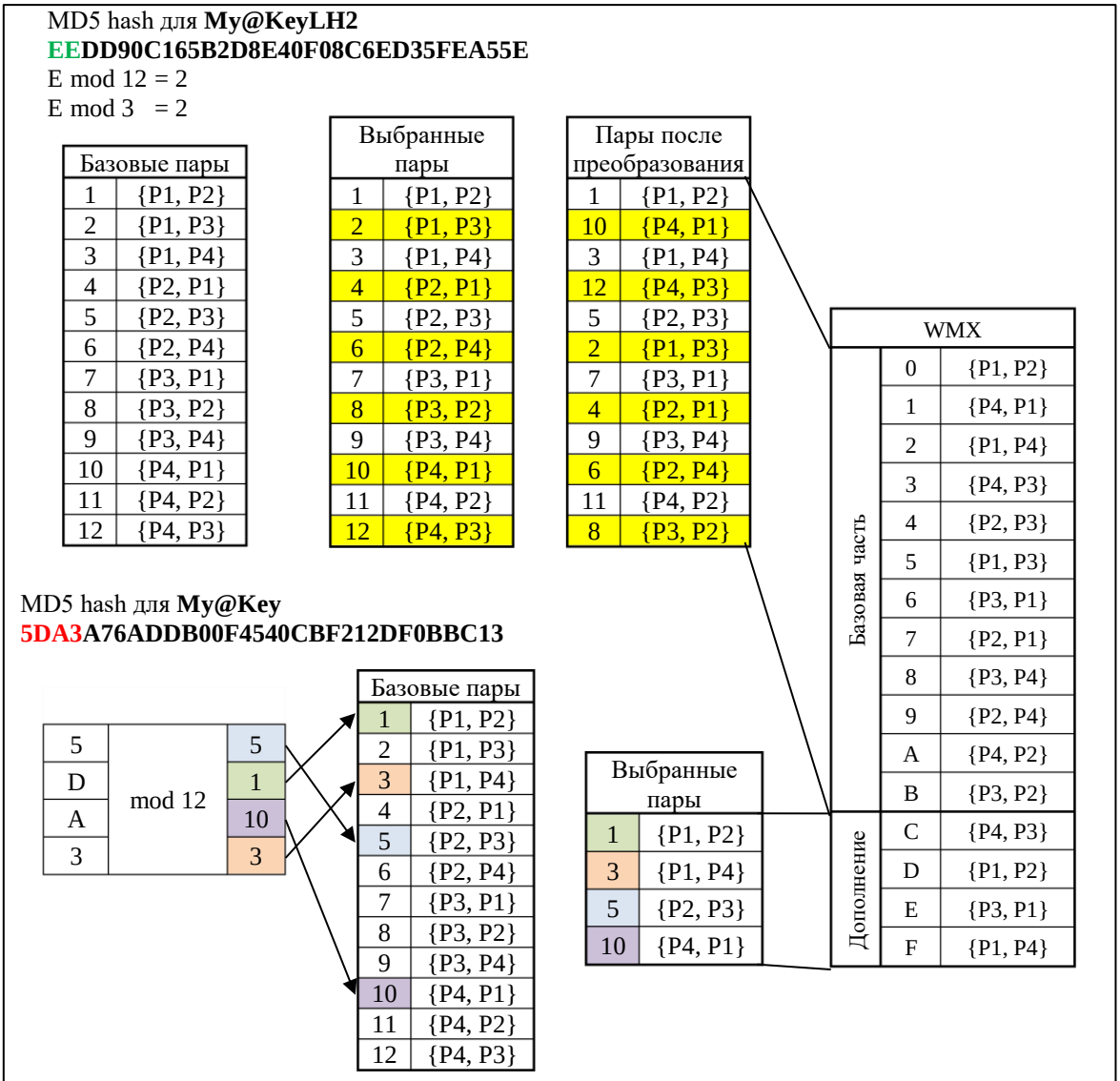


Рис. 6. Пример формирования таблицы WMX для ключа My@Key и частотной области LH2

Fig. 6. The example of WMX basis formation for My@Key key and LH2 frequency domain

чти на всех этапах: преобразование цветовых моделей, преобразование Арнольда, непосредственное встраивание битов ЦВЗ и т.п. Для параллельной реализации вычислений будет применяться распараллеливание цикла.

Последовательная реализация предполагает использование циклов следующего вида:

```
for(int i=0; i<size; i++)
{
    Обработка данных
}
```

Реализация с помощью OpenMP отличается добавлением специальной директивы:

```
#pragma omp parallel for
for(int i=0; i<size; i++)
{
    Обработка данных
}
```

При использовании данной директивы для разделения работы возможно использование опции *schedule*, которая будет выполнять балансировку нагрузки потоков (распределение итераций). Для того чтобы размер порции уменьшался с некоторого начального значения до величины *chunk* (по умолчанию *chunk* = 1), задается значение *guided*. В таком случае уменьшение порции будет пропорционально количеству еще не распределенных итераций, деленному на количество потоков, выполняющих обработку цикла. При этом количество итераций в последней порции может оказаться

меньше значения *chunk*. Форма записи директивы примет следующий вид:

```
#pragma omp parallel
for schedule(guided, chunk).
```

В большинстве случаев такое распределение позволяет аккуратнее разделить работу и сбалансировать загрузку потоков.

Экспериментальные исследования

Для экспериментов использованы по 100 фотографий каждого из разрешений: 1 280×720, 1 920×1 080, 2 560×1 440, 3 840×2 160 [18] (см. http://www.swsys.ru/uploaded/image/2021-3/2021-3-dop/1.jpg), а также компьютер с процессором Intel Core i7 4770. Технология Hyper-Threading была отключена. Максимальное дополнительное ускорение ядра (Turbo Boost) составляло 500 Mhz.

Выполняемые эксперименты можно разделить на три части:

– проверка ускорения трансформации Арнольда;

– проверка общего ускорения встраивания ЦВЗ при однопоточном вычислении;

– проверка общего ускорения встраивания ЦВЗ при использовании параллельных реализаций алгоритмов.

В первом эксперименте сравнивалось ускорение работы трансформации Арнольда с использованием таблиц преобразования *A1D* и *A2D* и без них. В этом эксперименте вычисления повторялись 10 000 раз, после чего находились средние значения. Визуализация результатов для блоков размером 32×32 пиксела (24 итерации – это возможный максимум) представлена на рисунке 7. По полученным результатам видно, что использование таблицы *A1D* позволяет получить среднее ускорение в 2,28 раза, в то время как таблица *A2D* преобразует блок за константное время. Для 24 итераций это дает ускорение в 30 раз. Для блоков большего размера (и, соответственно, для большего количества итераций) ускорение будет увеличиваться.

Во втором эксперименте проверялось общее ускорение внедрения ЦВЗ при вычислениях в одном потоке. В качестве передаваемой текстовой информации служил текст: *Some embedding text information*. В качестве секретного ключа использовалась строка, сформированная на основе базовой части Secret@Key#, к которой добавлялся порядковый номер эксперимента. Во время эксперимента дополнительно вычислялось время, затрачиваемое на каждый

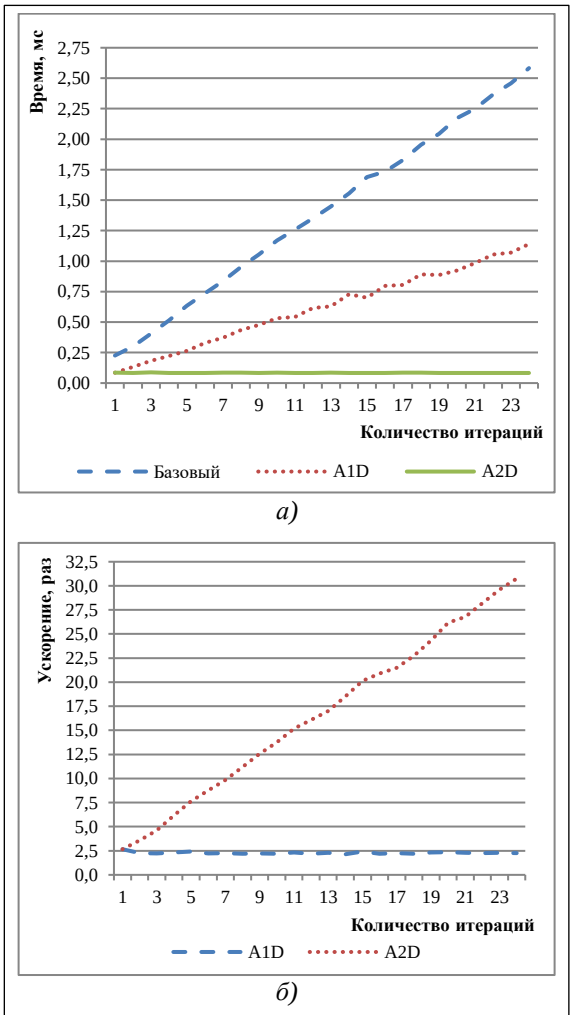


Рис. 7. Сравнение вычисления преобразования Арнольда в базовом методе и с использованием таблиц A1D и A2D для блока 32×32: а) время вычисления, б) ускорение

Fig. 7. Comparison of Arnold transform calculation in basic method (Base) and using tables A1D and A2D for 32×32 block: a) calculation time, б) speedup

шаг алгоритма встраивания ЦВЗ. Для получения более достоверных данных выполнялось по 1 000 замеров. Результаты, полученные для каждого из изображений, были усреднены и представлены в таблице 3. Согласно полученным данным, наибольшее ускорение от использования таблиц преобразования получил этап подготовки ЦВЗ, среднее ускорение составило 14,31 раза. Оценка ускорения подготовки ЦВЗ проведена для встраиваемой информации объемом 8–17 блоков. В случае встраивания битов ЦВЗ в частотные области было достигнуто ускорение в 1,28 раза для изображений 1 920×1 080.

Таблица 3

Сравнение базового и предлагаемого методов при однопоточном вычислении для изображений разрешением 1 920×1 080 пикселов

Table 3

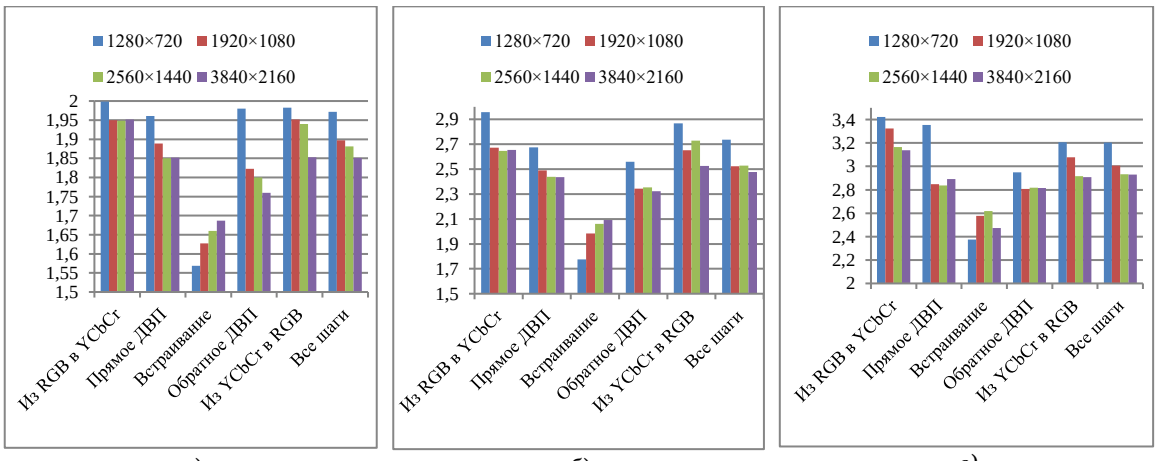
Comparison of base and proposed methods for single-threaded computation for 1 920×1 080 images

Шаг	Базовый метод, мс	Предложенный метод, мс	Коэффициент ускорения
Подготовка ЦВЗ	12,63	0,88	14,31
Из RGB в YCbCr	19,30	19,30	1,00
Прямое ДВП	15,28	14,69	1,04
Встраивание битов ЦВЗ	1,48	1,15	1,28
Обратное ДВП	17,31	16,80	1,03
Из YCbCr в RGB	15,43	15,43	1,00
Все шаги	81,45	68,28	1,19

Оценка влияния параллельных вычислений при подготовке ЦВЗ показала, что в среднем возможно достичь ускорения в 1,74 раза для двух потоков, при этом для трех и четырех потоков ускорение составило 2,54 и 3,24 раза соответственно.

В последнем эксперименте проверялось ускорение встраивания ЦВЗ при использовании параллельных вычислений на двух, трех и четырех потоках. В качестве передаваемой текстовой информации также использовалось сообщение *Some embedding text information*, в качестве секретного ключа – строка Secret@Key. Во время эксперимента дополнительно вычислялось время, затрачиваемое на каждый шаг алгоритма встраивания ЦВЗ, кроме подготовки. Вычисления проводились отдельно для изображений всех четырех разрешений, после чего были усреднены. В рамках экспериментов использовалось по 20 изображений для каждого разрешения и проводилось по 500 замеров времени. В таблице 4 показано среднее время вычисления для каждого типа разрешений на одном вычислительном ядре. Полученные коэффициенты ускорения при распараллеливании представлены в виде графиков (рис. 8). В среднем использование параллельных вычислений позволило достичь ускорения в 1,9 раза для двух потоков, для трех и четырех потоков ускорение составило 2,56 и 3,01 раза соответственно.

Можно заметить, что прирост производительности при использовании четырех потоков не очень большой, это обусловлено использованием в экспериментальных исследованиях четырехъядерного процессора (часть ресурсов используются операционной системой и фоновыми процессами). При этом следует учитывать, что в полученные результаты внесла свой вклад технология Turbo Boost, которая позволила делать более быстрые однопоточные вычисления. Использование параллельных вычислений дает дополнительное ускорение всего процесса встраивания в 1,9–3 раза в зависимости от количества потоков.



а) *б)* *в)*

Рис. 8. Сравнение ускорения шагов встраивания ЦВЗ при разном количестве вычислительных потоков: а) 2 потока, б) 3 потока, в) 4 потока

Fig. 8. Comparison of watermark embedding steps acceleration with a different number of computational threads: a) 2 threads, б) 3 threads, в) 4 threads

Таблица 4

Время встраивания ЦВЗ для изображений разных разрешений при однопоточном вычислении (мс)

Table 4

Execution time of the proposed method for different resolutions of images in single-threaded computation

Шаг	1 280×720	1 920×1 080	2 560×1 440	3 840×2 160
Из RGB в YCbCr	8,05	19,30	28,65	64,56
Прямое ДВП	6,12	14,69	22,07	50,92
Встраивание	0,54	1,15	1,72	3,88
Обратное ДВП	6,95	16,80	25,16	58,23
Из YCbCr в RGB	6,84	15,43	24,36	54,61
Все шаги	28,52	67,40	101,97	232,21

Заключение

В статье предложены несколько способов снижения вычислительных затрат и требуемого времени при подготовке и встраивании ЦВЗ с помощью преобразования Арнольда и ДВП. К таким способам относятся линейная интерпретация данных (носителя и ЦВЗ), использование таблиц преобразований и параллельные вычисления в нескольких вычислительных потоках. Предложенная двумерная таблица преобразования позволяет осуществлять скремблирование алгоритмом Арнольда за константное время. Для блоков ЦВЗ размером 32×32 пиксела это дает ускорение до 30 раз. Для изображений размером 1 920×1 080 было получено среднее ускорение этапа подготовки ЦВЗ в 14,31 раза и этапа встраивания битов ЦВЗ в 1,28 раза. Использование параллельных вычислений при встраивании ЦВЗ позволяет получить дополнительное ускорение в 1,90, 2,56 и 3,01 раза для двух, трех и четырех потоков соответственно. Рассмотренные способы дают возможность использовать технологии цифрового маркирования в мобильных технологиях с большей эффективностью.

Работа выполнена при поддержке РФФИ, проект № 19-07-00047 А.

Литература

1. Verma V., Jha R.K. An overview of robust digital image watermarking. IETE Technical Review, 2015, vol. 32, no. 6, pp. 479–496. DOI: 10.1080/02564602.2015.1042927.

2. Begum M., Uddin M.S. Digital image watermarking techniques: A review. Information, 2020, vol. 11, no. 2, p. 110. DOI: 10.3390/info11020110.

3. Abraham J., Paul V. An imperceptible spatial domain color image watermarking scheme. JKSUCI, 2019, vol. 31, no. 1, pp. 125–133. DOI: 10.1016/j.jksuci.2016.12.004.

4. Feng B., Li X., Jie Y., Guo C., Fu H. A novel semi-fragile digital watermarking scheme for scrambled image authentication and restoration. Mobile Networks and Applications, 2020, vol. 25, no. 1, pp. 82–94. DOI: 10.1007/s11036-018-1186-9.

5. Zeki A.M., Manaf A.A. A novel digital watermarking technique based on ISB (Intermediate Significant Bit). Intern. Scholarly and Scientific Research & Innovation, 2009, vol. 3, no. 2, pp. 444–451.

6. Mohammed G.N., Yasin A., Zeki A.M. Robust image watermarking based on Dual Intermediate Significant Bit (DISB). Intern. Journal of Digital Content Technology and its Applications, 2014, vol. 7, no. 5, pp. 18–22. DOI: 10.1109/CSIT.2014.6805973.

7. Gaata M.T. An efficient image watermarking approach based on Fourier transform. IJCA, 2016, vol. 136, no. 9, pp. 8–11. DOI: 10.5120/ijca2016908559.

8. Roy S., Pal A.K. A blind DCT based color watermarking algorithm for embedding multiple watermarks. AEU – International Journal of Electronics and Communications, 2017, vol. 72, pp. 149–161. DOI: 10.1016/J.AEUE.2016.12.003.

9. Khare P., Srivastava V.K. Secure and robust image watermarking scheme using homomorphic transform, SVD and Arnold transform in RDWT domain. Advances in Electrical and Electronic Engineering, 2019, vol. 17, no. 3, pp. 343–351. DOI: 10.15598/aeee.v17i3.3154.

10. Tan L., He Y., Wu F., Zhang D. A blind watermarking algorithm for digital image based on DWT. J. Phys.: Conf. Ser. Proc. CMVIT, 2020, vol. 1518, art. 012068. DOI: 10.1088/1742-6596/1518/1/012068.

11. Li Y., Wei M., Zhang F., Zhao J. A new double color image watermarking algorithm based on the SVD and Arnold scrambling. Journal of Applied Mathematics, 2016, vol. 2016, pp. 1–9. DOI: 10.1155/2016/2497379.

12. Zotin A., Favorskaya M., Proskurin A., Pakhirka A. Study of digital textual watermarking distortions under Internet attacks in high resolution videos. Procedia Computer Science, 2020, vol. 176, pp. 1633–1642. DOI: 10.1016/j.procs.2020.09.187.

13. Li M., Liang T., He Y. Arnold transform based image scrambling method. Proc. III ICMT-13, 2013, pp. 1309–1316. DOI: 10.2991/icmt-13.2013.160.

14. Ye R., Zhuang L. Baker map's itinerary based image scrambling method and its watermarking application in DWT domain. IJIGSP, 2012, vol. 4, no. 1, pp. 12–20. DOI: 10.5815/ijigsp.2012.01.02.

15. Yu X., Wang C., Zhou X. A survey on robust video watermarking algorithms for copyright protection. Applied Sciences, 2018, vol. 8, no. 10, art. 1891. DOI: 10.3390/app8101891.

16. Favorskaya M., Zotin A. Robust textual watermarking for high resolution videos based on Code-128 barcoding and DWT. Procedia Computer Science, 2020, vol. 176, pp. 1261–1270. DOI: 10.1016/j.procs.2020.09.135.

17. Slabaugh G., Boyes R., Yang X. Multicore image processing with OpenMP. IEEE Signal Processing Magazine, 2010, vol. 27, no. 2, pp. 134–138. DOI: 10.1109/MSP.2009.935452.

18. HD Wallpapers. URL: https://www.hdwallpapers.in/3840x2160_ultra+hd+4k-wallpapers-r.html (дата обращения: 12.03.2021).

Software & Systems Received 08.04.21
DOI: 10.15827/0236-235X.135.420-432 2021, vol. 34, no. 3, pp. 420–432

Methods for accelerating the preparation and embedding of a digital watermark using mobile devices based on Arnold and wavelet transforms

A.G. Zotin [1], *Ph.D. (Engineering), Associate Professor, zotin@sibsau.ru*
A.V. Proskurin [1], *Ph.D. (Engineering), Associate Professor, proskurin.av.wof@gmail.com*

[1] *Reshetnev Siberian State University of Science and Technology, Krasnoyarsk, 660037, Russian Federation*

Abstract. In recent years, digital watermarking technology has become increasingly important due to the explosive growth of data transmitted through unprotected communication channels. Digital watermarks can reduce the amount of transmitted information and be used to protect images (hosts) from illegal use. For a more effective implementation of the latter task, various studies are being carried out in order to improve robustness, imperceptibility and security of the watermark. In most cases, this leads to an increase in computational costs, which makes it difficult to use digital watermarking in mobile devices.

This work proposes several ways to reduce computational costs and computation time of digital watermarking methods based on Arnold and wavelet transforms. The first way consists in linear interpretation of digital watermark and a host, so it avoids the use of double cycles. The second way is to use lookup tables (LUT) to replace direct calculations. One of these tables allows performing the Arnold transform in certain time regardless of the number of iterations. Iterations for each block are determined using hash code of the secret key and specially formed tables. The third way of digital watermarks embedding time reduction is multithreaded execution implemented using the OpenMP technology. In combination with linear interpretation, this results in accelerations of 1.90, 2.56 and 3.01 times for two, three and four threads, respectively.

Keywords: digital watermarks, Arnold transform, Lookup tables, OpenMP.

Acknowledgements. *The work was financially supported by RFBR, project no. 19-07-00047 A.*

References

1. Verma V., Jha R.K. An overview of robust digital image watermarking. *IETE Technical Review*, 2015, vol. 32, no. 6, pp. 479–496. DOI: 10.1080/02564602.2015.1042927.

2. Begum M., Uddin M.S. Digital image watermarking techniques: A review. *Information*, 2020, vol. 11, no. 2, p. 110. DOI: 10.3390/info11020110.

3. Abraham J., Paul V. An imperceptible spatial domain color image watermarking scheme. *JKSUCI*, 2019, vol. 31, no. 1, pp. 125–133. DOI: 10.1016/j.jksuci.2016.12.004.

4. Feng B., Li X., Jie Y., Guo C., Fu H. A novel semi-fragile digital watermarking scheme for scrambled image authentication and restoration. *Mobile Networks and Applications*, 2020, vol. 25, no. 1, pp. 82–94. DOI: 10.1007/s11036-018-1186-9.

5. Zeki A.M., Manaf A.A. A novel digital watermarking technique based on ISB (Intermediate Significant Bit). *Intern. Scholarly and Scientific Research & Innovation*, 2009, vol. 3, no. 2, pp. 444–451.

6. Mohammed G.N., Yasin A., Zeki A.M. Robust image watermarking based on Dual Intermediate Significant Bit (DISB). *Intern. Journal of Digital Content Technology and its Applications*, 2014, vol. 7, no. 5, pp. 18–22. DOI: 10.1109/CSIT.2014.6805973.

7. Gaata M.T. An efficient image watermarking approach based on Fourier transform. *IJCA*, 2016, vol. 136, no. 9, pp. 8–11. DOI: 10.5120/ijca2016908559.

8. Roy S., Pal A.K. A blind DCT based color watermarking algorithm for embedding multiple watermarks. *AEU – International Journal of Electronics and Communications*, 2017, vol. 72, pp. 149–161. DOI: 10.1016/J.AEUE.2016.12.003.

9. Khare P., Srivastava V.K. Secure and robust image watermarking scheme using homomorphic transform, SVD and Arnold transform in RDWT domain. *Advances in Electrical and Electronic Engineering*, 2019, vol. 17, no. 3, pp. 343–351. DOI: 10.15598/aeee.v17i3.3154.

10. Tan L., He Y., Wu F., Zhang D. A blind watermarking algorithm for digital image based on DWT. *J. Phys.: Conf. Ser. Proc. CMVIT*, 2020, vol. 1518, art. 012068. DOI: 10.1088/1742-6596/1518/1/012068.

11. Li Y., Wei M., Zhang F., Zhao J. A new double color image watermarking algorithm based on the SVD and Arnold scrambling. *Journal of Applied Mathematics*, 2016, vol. 2016, pp. 1–9. DOI: 10.1155/2016/2497379.

12. Zotin A., Favorskaya M., Proskurin A., Pakhirka A. Study of digital textual watermarking distortions under Internet attacks in high resolution videos. *Procedia Computer Science*, 2020, vol. 176, pp. 1633–1642. DOI: 10.1016/j.procs.2020.09.187.

13. Li M., Liang T., He Y. Arnold transform based image scrambling method. *Proc. III ICMT-13*, 2013, pp. 1309–1316. DOI: 10.2991/icmt-13.2013.160.

14. Ye R., Zhuang L. Baker map's itinerary based image scrambling method and its watermarking application in DWT domain. *IJIGSP*, 2012, vol. 4, no. 1, pp. 12–20. DOI: 10.5815/ijigsp.2012.01.02.

15. Yu X., Wang C., Zhou X. A survey on robust video watermarking algorithms for copyright protection. *Applied Sciences*, 2018, vol. 8, no. 10, art. 1891. DOI: 10.3390/app8101891.

16. Favorskaya M., Zotin A. Robust textual watermarking for high resolution videos based on Code-128 barcoding and DWT. *Procedia Computer Science*, 2020, vol. 176, pp. 1261–1270. DOI: 10.1016/j.procs.2020.09.135.

17. Slabaugh G., Boyes R., Yang X. Multicore image processing with OpenMP. *IEEE Signal Processing Magazine*, 2010, vol. 27, no. 2, pp. 134–138. DOI: 10.1109/MSP.2009.935452.

18. *PB Wallpapers*. Available at: https://www.hdwallpapers.in/3840x2160_ultra+hd+4k-wallpapers-r.html (accessed March 12, 2021).

Для цитирования

Зотин А.Г., Проскурин А.В. Способы ускорения подготовки и встраивания цифрового водяного знака с использованием мобильных устройств на основе преобразования Арнольда и вейвлет-преобразования // Программные продукты и системы. 2021. Т. 34. № 3. С. 420–432. DOI: 10.15827/0236-235X.135.420-432.

For citation

Zotin A.G., Proskurin A.V. Methods for accelerating the preparation and embedding of a digital watermark using mobile devices based on Arnold and wavelet transforms. *Software & Systems*, 2021, vol. 34, no. 3, pp. 420–432 (in Russ.). DOI: 10.15827/0236-235X.135.420-432.