

Компрессия изображения с использованием теории вейвлетов на примере изображения штампованной детали*

В. З. Рахманкулов, А. А. Ахрем, А. А. Барсегов,
В. В. Герасимов, В. В. Лебедев

В статье рассматриваются теоретические и практические аспекты сжатия изображений с использованием теории вейвлетов. Представлены алгоритм и пример компрессии изображений.

1. Введение

Одной из основных задач при работе с компьютерной графикой и со статическими изображениями — это наиболее точное и четкое представление графического объекта (цифрового изображения) при минимальных затратах аппаратных ресурсов на это представление. Как правило, для качественной обработки цифровых изображений или даже обычного просмотра требовались довольно мощные аппаратные возможности технических средств — будь то ПК (персональный компьютер) или СТЗ (система технического зрения) роботов. На первый план в решении данной проблемы выходит поиск наиболее эффективных алгоритмов и методов по компрессии и декомпрессии изображений для последующей их программной реализации и использования на практике. Помимо этого данные алгоритмы должны обладать достаточной гибкостью и универсальностью, чтобы существовала возможность регулирования степени

* Работа выполнена при поддержке программы Президиума РАН «Фундаментальные проблемы информатики и информационных технологий» (проект № 2.44).

сжатия. Для решения подобной проблемы могут быть использованы вейвлеты, по своим свойствам, подходящие для применения в компрессии изображений (сигналов). Идея вейвлет-сжатия также напоминает идею сжатия, основанную на использовании преобразования Фурье. Дискретное преобразование Фурье ставит в соответствие набору из N значений функции, выражающей, например, зависимость некоторого параметра от пространственной координаты, набор из N коэффициентов Фурье — элементов частотного спектра этой функции. Если некоторые коэффициенты Фурье оказались равными 0, то оставшихся значений будет достаточно, чтобы полностью (без искажений) восстановить исходный набор. Таким образом, удастся представить исходную информацию меньшим числом значений. Дополнительно можно установить правила исключения ненулевых коэффициентов Фурье, например, исключать коэффициенты, модули которых окажутся меньше заданного числа. Разумеется, это приведет к искажению информации, но позволит увеличить степень сжатия [1].

Понятно, что хорошую степень сжатия с малыми потерями информации можно получить лишь в случае, когда многие коэффициенты Фурье близки либо равны нулю. Реальные изображения имеют весьма сложную структуру, и их Фурье-образ может не удовлетворять подобным требованиям. Можно разбить изображения на области фиксированного размера и выполнять преобразования в каждой области отдельно. В каждой такой области изображение будет иметь меньше особенностей, чем все изображение в целом, и образы Фурье этих областей могут оказаться более подходящими для сжатия, чем образ всего изображения. Именно этот принцип был положен в основу создания популярного графического формата JPEG. Недостатком такого подхода является то, что при высокой степени сжатия части единого изображения, обрабатываемые независимо друг от друга, могут плохо стыковаться — становится видно, что изображение собрано из разных кусков.

Дискретное вейвлет-преобразование также переводит набор из N элементов в набор из N вейвлет-коэффициентов. Эти коэффициенты соответствуют не только амплитудам различных частот, но и различным пространственным участкам изображения. Таким образом, отпадает необходимость в предварительном разбиении изображения на локальные области — эта возможность уже заложена в вейвлет-преобразовании, причем при этом не возникает никаких нежелательных побочных эффектов [2].

2. Алгоритм компрессии изображений с использованием вейвлет-преобразования

Цифровое изображение — это массив значений пикселей, который мы можем представить в виде списка чисел. Задача сжатия состоит из двух основных частей: кодирования и декодирования. Кодирование представляет исходный список чисел иначе, чтобы сократить требуемый для массива объем памяти. Декодирование восстанавливает исходное изображение из закодированного. Если декодированное изображение в точности соответствует исходному, то алгоритм кодирования — декодирования называется алгоритмом без потерь. Если декодированное изображение отличается от исходного изображения, то алгоритм называется алгоритмом с потерями.

На рис. 1 представлена схема процесса сжатия изображения. Идея метода состоит в том, чтобы преобразовать данные изображения к другому виду, где легче определить часть данных, которую можно было бы безболезненно удалить. Это позволяет отбросить значительную часть данных с небольшой потерей качества изображения. Для вейвлет-преобразования это данные, соответствующие мелким деталям. Результат алгоритмического кодирования может быть затем сжат путем компактного представления закодированных чисел. Дискретизация (квантование) может дать как уменьшение количества чисел в списке, так и уменьшение количества бит, необходимых для записи каждого числа. При декодировании исходное изображение восстанавливается из закодированных данных [3].

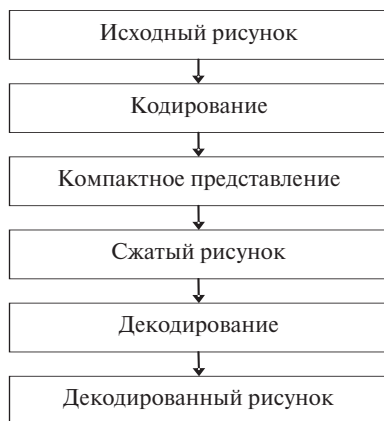


Рис. 1. Схема процесса сжатия изображения

Примером вейвлета является функция Хаара:

$$H(t) \equiv \begin{cases} 1, & \text{если } t \in \left[0, \frac{1}{2}\right); \\ -1, & \text{если } t \in \left[\frac{1}{2}, 1\right]; \\ 0, & \text{если } t \notin [0, 1]. \end{cases}$$

На рис. 2 приведен график функции Хаара.

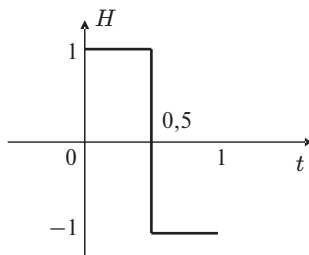


Рис. 2

На рис. 3 приведен пример сжатия изображения в Wavelets Toolbox с использованием вейвлета Хаара. Вейвлет-сжатие изображений состоит в том, что сначала к изображению применяется вейвлет-преобразование, а затем из данных преобразованного изображения удаляются некоторые коэффициенты. К оставшимся коэффициентам может быть применено кодирование. Сжатое изображение восстанавливается путем декодирования коэффициентов, если это необходимо, и применением обратного преобразования к результату [3].

На рис. 4 приведены изображения детали до и после сжатия. В примере приведенном на рис. 2 был выбран в качестве базисного вейвлет Хаара. В результате разложения до 2-го уровня изображение сжимается в 4 раза с сохранением 96,64 % энергии. Удалено 93,43 % вейвлет коэффициентов, при заданном числовом пороге 40 единиц. При этом полезная информация в изображении не теряется.

3. Вейвлет-преобразование

Усреднение и выделение деталей при помощи вейвлета Хаара можно представить, как применение низкочастотных и высокочастотных фильтров.

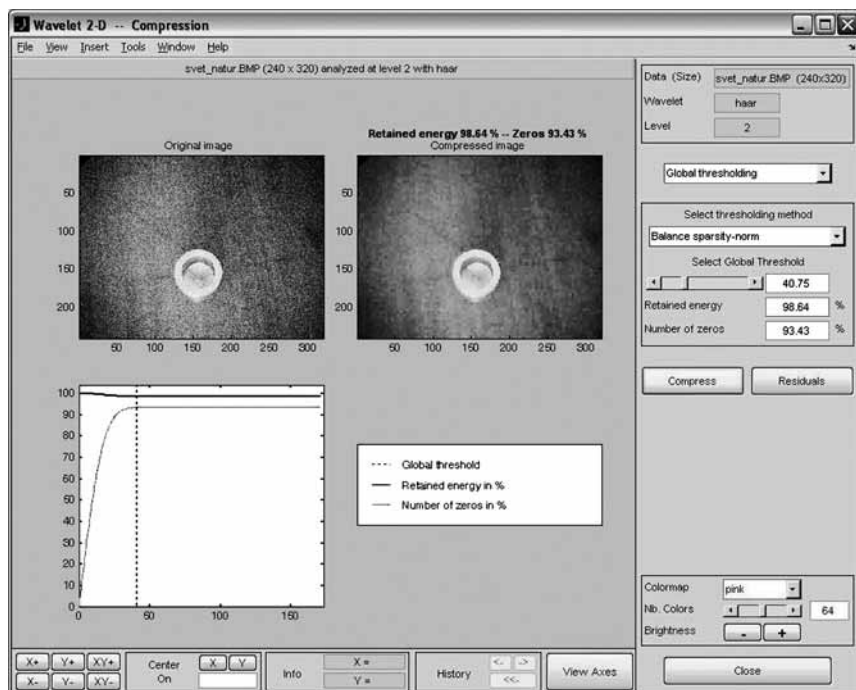


Рис. 3. Пример сжатия изображения в Wavelets Toolbox с использованием вейвлета Хаара

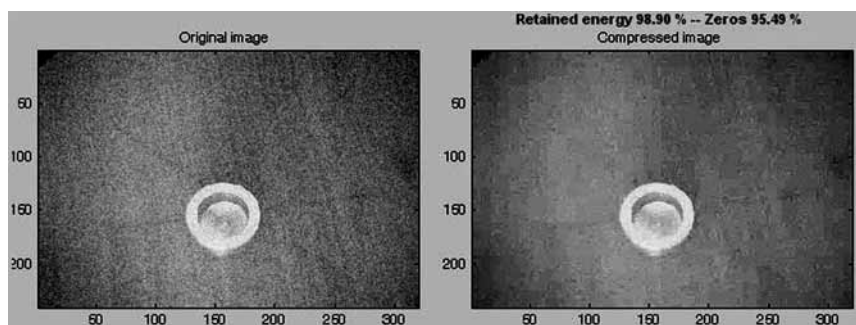


Рис. 4. Изображение детали до и после сжатия

Для ряда задач, например, для обработки растровых изображений, необходимо обрабатывать двумерные массивы данных. Многомасштабный анализ должен строиться в пространстве функций двух переменных.

Самый простой и широко распространенный путь — тензорное произведение одномерных масштабных анализов (МА). В качестве двумерной скейлинг-функции берется

$$\Phi(t, s) = \varphi(t)\varphi(s).$$

Вместо одного вейвлета возникает три:

$$\Phi_{LH}(t, s) = \varphi(t)\varphi(s), \Phi_{HL}(t, s) = \varphi(t)\varphi(s), \Phi_{HH}(t, s) = \varphi(t)\varphi(s).$$

(L означает низкую частоту, H — высокую частоту). Пространства V_j порождаются сдвигами скейлинг — функции на одном и том же масштабе:

$$V_J = \{2^j \Psi_L H(2^j t - k, 2^j s - m)\},$$

пространства деталей имеют вид:

$$W_j^{LH} = \{2^j \Psi_{LH}(2^j t - k, 2^j s - m)\},$$

$$W_j^{HL} = \{2^j \Psi_{LH}(2^j t - k, 2^j s - m)\},$$

$$W_j^{HH} = \{2^j \Psi_{LH}(2^j t - k, 2^j s - m)\}.$$

Соответствующие проекции находятся применением фильтров (в очевидных обозначениях):

$$x \mapsto \{H_t \otimes H_s(x), H_t \otimes G_s(x), G_t \otimes H_s(x), G_t \otimes G_s(x)\},$$

где x — двумерный сигнал.

На рис. 5, рис. 6 представлена графическая интерпретация двумерного вейвлет-преобразования. Двумерное вейвлет-преобразование означает применение одномерного вейвлет-преобразования к каждой строке, а затем к каждому столбцу информации.

Четверть коэффициентов двумерного преобразования является результатом действия высокочастотного фильтра (H) на строки, а затем на столбцы. Этот блок коэффициентов в правом нижнем углу назовем

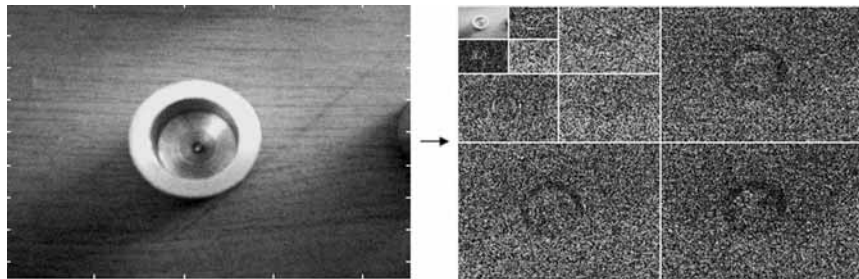


Рис. 5. Двумерное вейвлет-преобразование

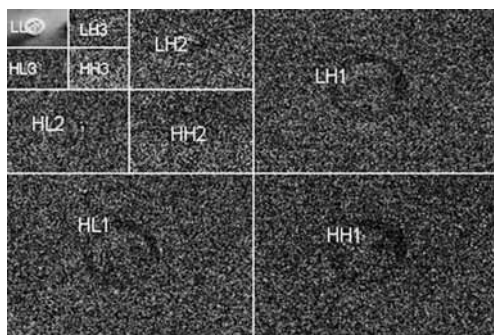


Рис. 6. Схема двумерного вейвлет-преобразования

HH1. Другая четверть — результат действия низкочастотного оператора (*L*) на столбцы, которые уже прошли через *H*. Этот блок в правом верхнем углу назовем *LH1*. Меньшие блоки показывают, что операции вейвлет-преобразования последовательно воздействуют на все меньшее количество коэффициентов [4].

После вейвлет-преобразования вейвлет-коэффициенты сортируются. Запоминаются только *X* наибольших коэффициентов, а оставшиеся наименьшие коэффициенты полагаются равными нулю. Значение *X* выбирается пользователем. Декодирование производится путем применения обратного вейвлет преобразования к прореженному массиву коэффициентов.

4. Нуль-дерева вейвлетов

Структурируем блоки вейвлет-преобразования в дерево, при этом каждый коэффициент в блоке имеет дочерние коэффициенты в блоках следующего уровня (см. рис. 7).

Смысл введения структуры дерева в том, что коэффициенты в блоках на различных уровнях обладают значительной степенью подобия. Дочерние коэффициенты будут похожи на своих родителей, находящихся на предыдущем уровне [1].

5. Вейвлет-кодирование с нуль-деревом

Коэффициент является значимым, если его абсолютное значение больше пороговой величины. Поскольку вейвлет-коэффициенты будут в двоичном виде, то пороговое значение также задаем степенью двойки.

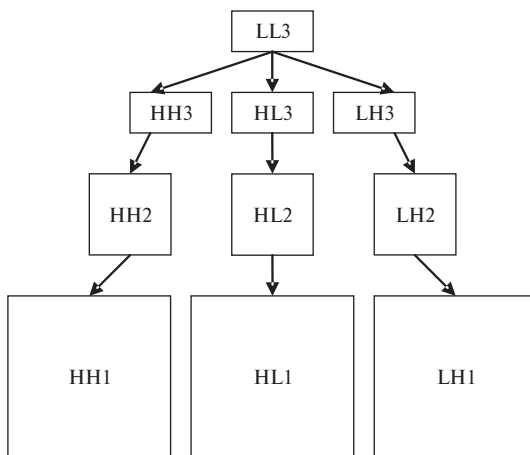


Рис. 7. Нуль-дерево

Создадим массив того же размера что и изображение, чтобы отслеживать этап обработки. Вначале цикла обнулим его, чтобы показать, что элементы еще не обрабатывались.

Низкочастотные *LL* элементы сосредоточены в области наибольших по модулю коэффициентов. Запишем их значения, не кодируя.

Чтобы обозначить обрабатываемый элемент, используем обозначения:

POS: вейвлет-коэффициент после прохождения порога является значимым и положительным.

NEG: вейвлет-коэффициент после прохождения порога является значимым и отрицательным.

IZ: вейвлет-коэффициент после прохождения порога является незначимым, но имеет значимых потомков в поддереве под ним (*изолированный ноль*).

ZR: вейвлет-коэффициент после прохождения порога является незначимым, и все поддерево потомков этого коэффициента содержит только незначимые коэффициенты (*корень нуля дерева*).

ZT: незначимый потомок коэффициента, являющегося корнем нуля дерева.

Если абсолютное значение коэффициента превышает порог, то записываем значение POS или NEG. Чтобы классифицировать незначимые коэффициенты, нужна информация о коэффициентах, расположенных выше и ниже текущего. Для этого делаем два прохода: снизу вверх (чтобы обнаружить родителей значимых потомков), и сверху вниз (чтобы обнару-



Рис. 8. Блок-схема алгоритма «проход снизу-вверх»

жить корни нуль дерева) [1, 2]. На рис. 8 приведена блок-схема алгоритма «проход снизу-вверх».

В конце прохода снизу вверх все значимые коэффициенты и изолированные нули будут определены и закодированы. Чем являются оставшиеся коэффициенты (ZR или ZT), определим с помощью прохода сверху вниз (см. рис. 9). Чтобы классифицировать незначимые коэффициенты, нужна информация о коэффициентах, расположенных выше и ниже текущего.

Теперь объединим все части алгоритма (см. рис. 10).

6. Декодирование изображения

Как и в кодировании, для хранения символов используем массив такого же размера, как и изображение. Массив инициализируется нулями при каждом проходе (при изменении порога). Для каждого порогового значения мы считываем символы для поддиапазонов *LH*, *HH* и *HL* сверху вниз. Каждый раз, когда встречается символ ZR, он распространяется на все поддерево путем сопоставления символа ZT всем его потомкам [1].

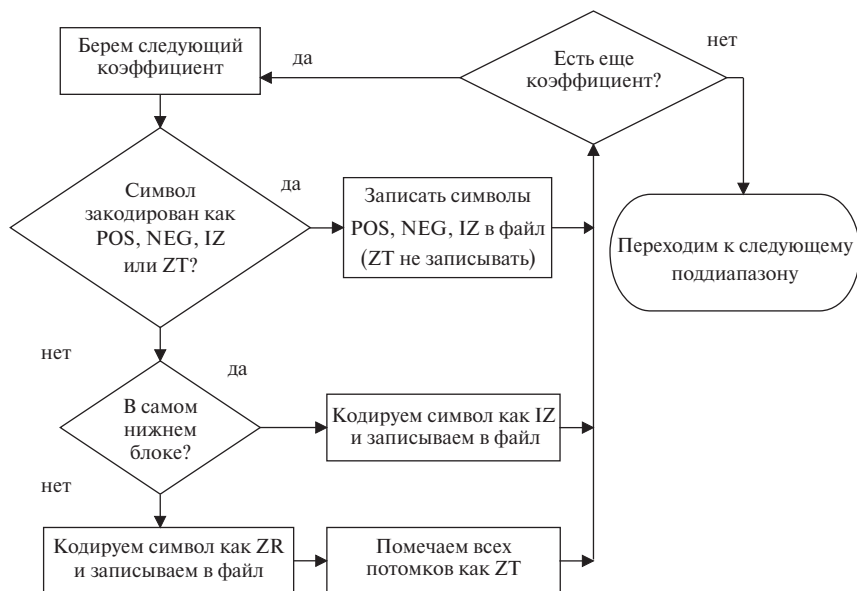


Рис. 9. Блок-схема алгоритма «проход сверху-вниз»

Значения коэффициентов вейвлет преобразования реконструируются по их двоичным представлениям, причем вначале задействуются наиболее значимые биты (Коэффициент является значимым, если его абсолютное значение больше пороговой величины), а затем менее значимые. Декодированное изображение получают путем применения обратного вейвлет преобразования, после того как построена последняя битовая плоскость [2] (см. рис. 11).

Заключение

Как было показано в примере компрессии изображения детали с использованием Wavelets Toolbox, результат сжатия напрямую зависит как от выбора базисного вейвлета, так и выбора значения порога при ограничении уровня детализирующих коэффициентов.

Необходимо помнить, что если отбросить коэффициенты, значения которых близки к нулю, возможно, произойдет необратимая потеря информации, ведь отброшенные коэффициенты участвуют в формировании изображения. Поэтому выбранное пороговое значение коэффициентов сильно влияет на качество изображения — задание слишком высокого порога повлечет за собой падение качества. Насколько будет оптимальным

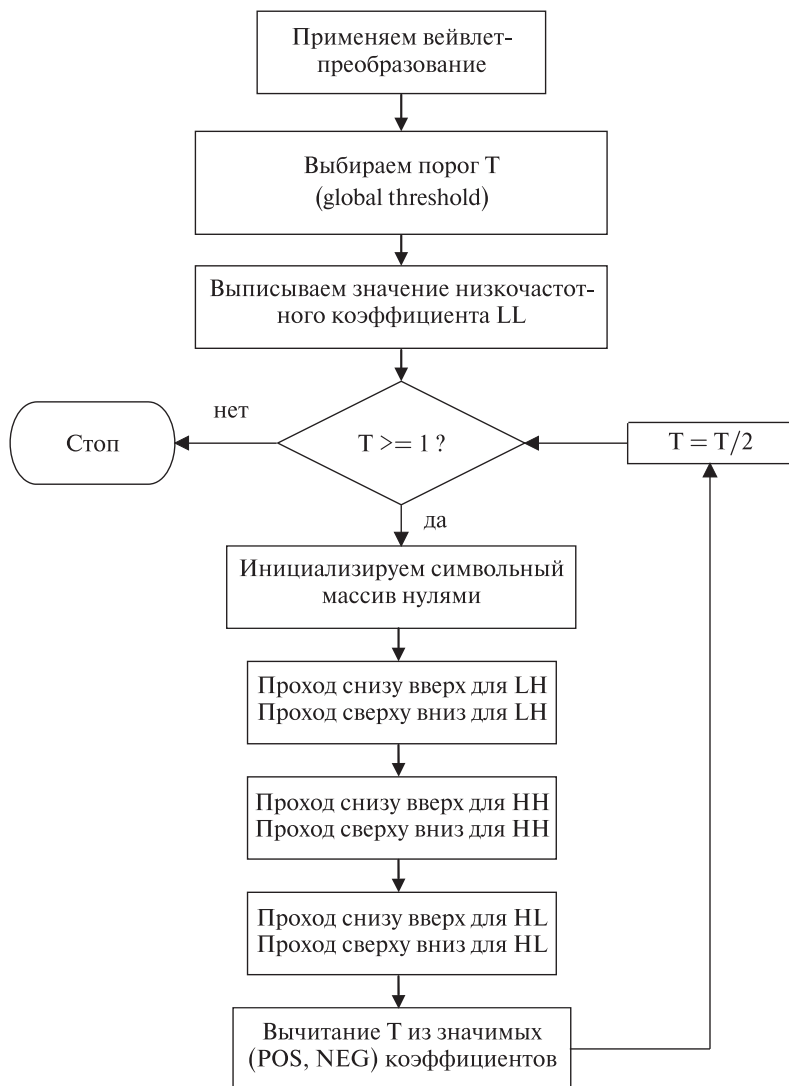


Рис. 10. Блок-схема алгоритма кодирования с нуль-деревом

этот выбор зависит прежде всего от конкретного изображения и решаемых целей при его сжатии. Поэтому возможность выбирать между высокой степенью сжатия и потерей информации предоставляется конечному пользователю.

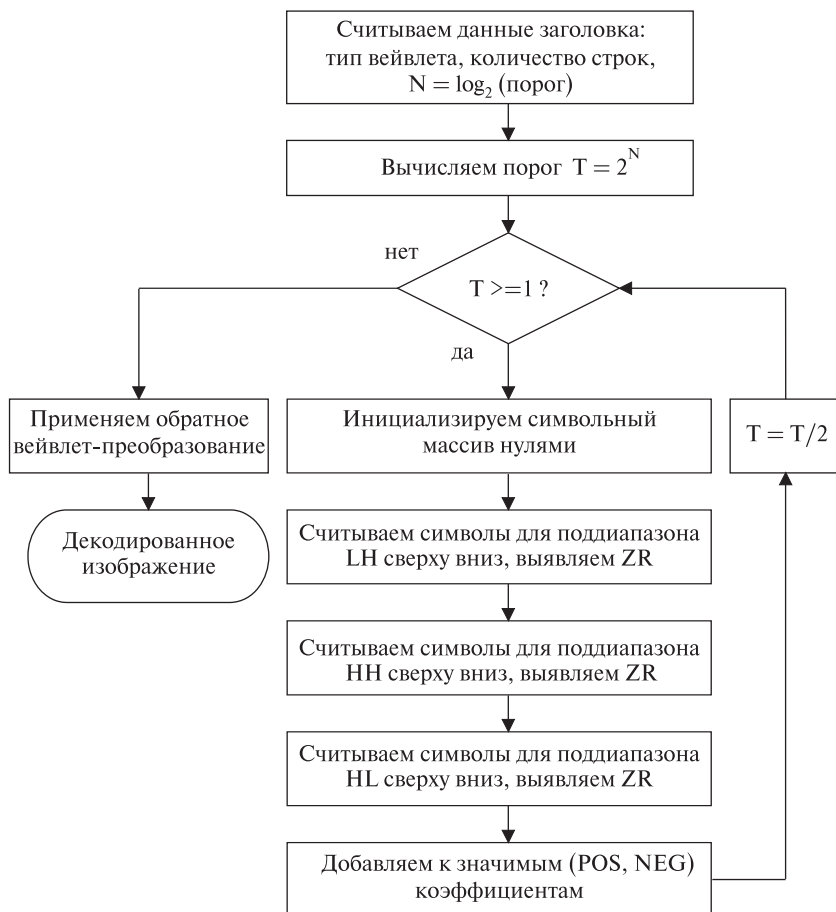


Рис. 11. Блок-схема алгоритма декодирования с нуль-деревом

Литература

1. Поршнев С. В. Оценка возможности использования непрерывного вейвлет-преобразования для обработки широкополосного частотно-модулированного сигнала. [Электронный ресурс] <http://jre.cplire.ru/koi/jan03/3/text.html>
2. Учебный курс по вейвлет-анализу. [Электронный ресурс] http://algotlist.manual.ru/compress/image/leo_lev/.
3. Дьяконов В. П. Вейвлеты. От теории к практике М.: СОЛОН-Р, 2004.
4. Малла С. Вейвлеты в обработке сигналов. Пер. с англ. М.: Мир, 2005.