

# Оптимизация алгоритмов сжатия цифровых сигналов на системах с общей памятью

А.В. Данилова<sup>1</sup>, Н.В. Пауков<sup>1</sup>, М.А. Теплякова<sup>1</sup>

<sup>1</sup>Воронежский государственный университет

Информационные технологии в настоящее время стремительно развиваются и активно внедряются в различные области науки и техники. Одним из перспективных направлений является цифровая обработка сигналов. Для хранения и передачи информации используются алгоритмы сжатия. При этом для различных видов сигналов более эффективными оказываются разные методы. В данной работе мы рассматриваем задачу поиска оптимального алгоритма сжатия, соответствующего сигналам каждого конкретного типа. С этой целью мы разработали алгоритм перебора преобразований сигнала, для реализации которого используются ресурсы суперкомпьютерного центра ВГУ.

*Ключевые слова:* сжатие сигналов, параллельное программирование, дискретное преобразование Фурье.

## 1. Введение

В настоящее время создание новых методов и алгоритмов обработки цифровых сигналов является перспективной и крайне важной областью научных исследований. Первой ключевой задачей в этом направлении является очистка сигнала от постороннего шума, появляющегося в результате неточных измерений, фоновых помех или неисправности аппаратуры. Во-вторых, большую практическую значимость имеют методы сжатия, основная идея которых в том, чтобы сохранить всю важную информацию, присутствующую в сигнале, сократив при этом объём памяти, используемый для её хранения.

Многие способы сжатия основаны на использовании дискретного преобразования Фурье (ДПФ), поскольку оно оказывается эффективным для широкого спектра сигналов [1, 2]. ДПФ используется в реализации многих методов сжатия изображения и видео, таких как JPEG, MPEG и прочих. Также в настоящее время для цифровой обработки сигналов активно используются вейвлеты различных типов [3, 4].

Существенным аспектом является тот факт, что для сигналов разной структуры эффективнее оказываются те или иные конкретные методы. Следовательно, разработка алгоритмов автоматического отыскания оптимального способа сжатия для определенного типа сигналов представляет серьёзный практический интерес. Таким образом, целью нашего исследования мы решили поставить создание подобного алгоритма.

При наличии значительных вычислительных ресурсов, указанную выше задачу можно было бы решить простым перебором различных способов преобразования сигнала, но в реальных условиях это заняло бы слишком большое количество времени. Помимо этого, немаловажно чётко определить то, какие методы сжатия мы будем считать оптимальными. Таким образом, что касается конкретных задач, то стоит отметить следующие:

- выбор критерия для определения оптимальности конкретных способов сжатия;
- создание эффективного алгоритма для поиска оптимальных методов сжатия.

Так как задачи оптимизации с большими объемами данных являются весьма ресурсозатратными, то для реализации поставленной цели было принято решение использовать алгоритмы параллельного программирования для систем с общей памятью. Для реализации вычислительных алгоритмов был выбран один из самых распространенных стандартов программирования систем с общей памятью OpenMP. Расчеты производились с использованием ресурсов Суперкомпьютерного центра Воронежского государственного университета.

## 2. Методика расчётов

Рассмотрим цифровой сигнал как некоторый вектор  $F$ . Тогда, как правило, все методы сжатия устроены следующим образом.

### 2.1. Анализ

К рассматриваемому сигналу применяется некоторое невырожденное линейное преобразование

$$G = T^{-1}F \quad (1)$$

где  $G$  - преобразованный сигнал,  $T$  - матрица преобразования,  $T^{-1}$  - обратная матрица. В дальнейшем мы будем использовать краткое обозначение  $S = T^{-1}$ .

### 2.2. Непосредственно сжатие

Сигнал  $G$  подвергают каким-либо изменениям. В задачах сжатия обычно наименьшие по модулю компоненты  $G$  полагают равными нулю. Полученный после этого вектор  $\tilde{G}$  сохраняют, и он занимает в памяти меньше места, чем  $G$ . Процентное отношение количества компонент, которые были отброшены, к общей длине вектора  $G$  мы будем называть уровнем сжатия.

### 2.3. Синтез

К вектору  $\tilde{G}$  применяется обратное преобразование

$$\tilde{F} = T\tilde{G} \quad (2)$$

где  $\tilde{F}$  – восстановленный сигнал. Очевидно, если сжатие не производилось  $G = \tilde{G}$ , то восстановленный сигнал (без учета погрешности вычислений) совпадает с исходным  $F = \tilde{F}$ . В общем случае  $F$  и  $\tilde{F}$  будут отличаться, и допустимым является уровень сжатия, при котором их различия не будут затрагивать полезную часть информации.

Приемлемое значение отклонения восстановленного сигнала от исходного зависит от конкретной задачи, которую решает исследователь. В данной работе для оценки степени соответствия  $F$  и  $\tilde{F}$  мы применим широко используемый общий параметр – отклонение по среднеквадратичной норме  $\sigma$ :

$$\sigma = \|F - \tilde{F}\| = \sqrt{\frac{\sum_{k=0}^n (f_k - \tilde{f}_k)^2}{n+1}} \quad (3)$$

где  $n+1$  – длина векторов  $F$  и  $\tilde{F}$ ,  $f_k$  и  $\tilde{f}_k$  – их компоненты. Оптимальной будем считать матрицу преобразования  $S$ , имеющую при заданном уровне сжатия среднюю ошибку  $\sigma$  меньшую, чем ДПФ, или же близкую к нему.

Для расчетов мы используем дискретное косинусное преобразование Фурье, которое задается следующим образом [5]

$$\hat{f}_m = \frac{1}{\sqrt{2n+1}}(f_0 + 2 \sum_{k=1}^n f_k \cos(\frac{2\pi km}{2n+1})), m = 0, 1, \dots, n. \quad (4)$$

Формула обращения тогда имеет вид

$$f_k = \frac{1}{\sqrt{2n+1}}(\hat{f}_0 + 2 \sum_{m=1}^n \hat{f}_m \cos(\frac{2\pi km}{2n+1})), k = 0, 1, \dots, n. \quad (5)$$

Данному преобразованию соответствует следующая матрица

$$S = \frac{1}{\sqrt{2n+1}} \begin{pmatrix} 1 & 2 & 2 & \dots & 2 \\ 1 & 2 \cos \frac{2\pi}{2n+1} & 2 \cos \frac{4\pi}{2n+1} & \dots & 2 \cos \frac{2\pi n}{2n+1} \\ 1 & 2 \cos \frac{4\pi}{2n+1} & 2 \cos \frac{8\pi}{2n+1} & \dots & 2 \cos \frac{4\pi n}{2n+1} \\ \vdots & \vdots & \ddots & \vdots & \\ 1 & 2 \cos \frac{2\pi n}{2n+1} & 2 \cos \frac{4\pi n}{2n+1} & \dots & 2 \cos \frac{2\pi n^2}{2n+1} \end{pmatrix} \quad (6)$$

Как видим, в данном случае  $S = T^{-1} = T$

### 3. Структура алгоритма поиска оптимального преобразования

Ключевым этапом алгоритма поиска оптимального метода сжатия является генерация исследуемого набора матриц преобразования. Во время поиска оптимальной матрицы среди исследуемого набора возникает необходимость в хранении информации о рассматриваемых матрицах. При этом нужно учитывать аппаратные ограничения используемого для расчётов вычислительного устройства, так как в данном случае особенности работы с памятью существенно влияют на построение алгоритма. С целью уменьшения количества матриц, одновременно хранящихся в памяти устройства, был рассмотрен следующий вариант алгоритма генерации матриц преобразования и поиска наиболее эффективной матрицы.

В качестве возможных значений элементов матрицы были выбраны числа 1,  $-1$  и 0. При этом максимальное количество матриц, полученное вариацией значений элементов, было бы равно  $3^{(n+1)^2}$ . Для уменьшения объема вычислений было решено организовать строгую классификацию генерируемых матриц, которая бы позволила избежать одновременного хранения всего исследуемого спектра матриц. Данная классификация должна отражать степень эффективности матриц преобразования, а также их элементный состав. Было решено разбить матрицы на классы по количеству входящих в них элементов, имеющих значение 1. После нахождения наиболее оптимального класса матриц, его также разбить на подклассы по количеству элементов, имеющих значение  $-1$ .

Но даже после выделения этой группы матриц, предположительно обладающих высокой эффективностью для сжатия рассматриваемой группы сигналов, выбранные матрицы всё ещё являются слишком разнородными и, следовательно, нельзя утверждать, что они обладают одинаковой эффективностью сжатия. Для того чтобы разрешить эту проблему, можно провести дальнейшую классификацию матриц, учитывающую количество элементов со значениями 1 и  $-1$  по строкам, а также их взаимное месторасположение.

Однако подобное разбиение матриц преобразования на классы не позволяет учесть случайных факторов, влияющих на эффективность преобразования. Кроме того, нельзя утверждать, что матрицы, составленные из элементов, имеющих ограниченный дискретный набор значений, подходят для решения поставленной задачи. Вследствие этого было решено отказаться от описанного ранее алгоритма. Вместо него использовался представленный далее алгоритм, созданный с учётом вышеозначенных идей.

### 4. Алгоритм поиска оптимального преобразования

Мы предлагаем построить расчеты по итерационному принципу. Схема предложенного нами алгоритма показана на рис. 1.

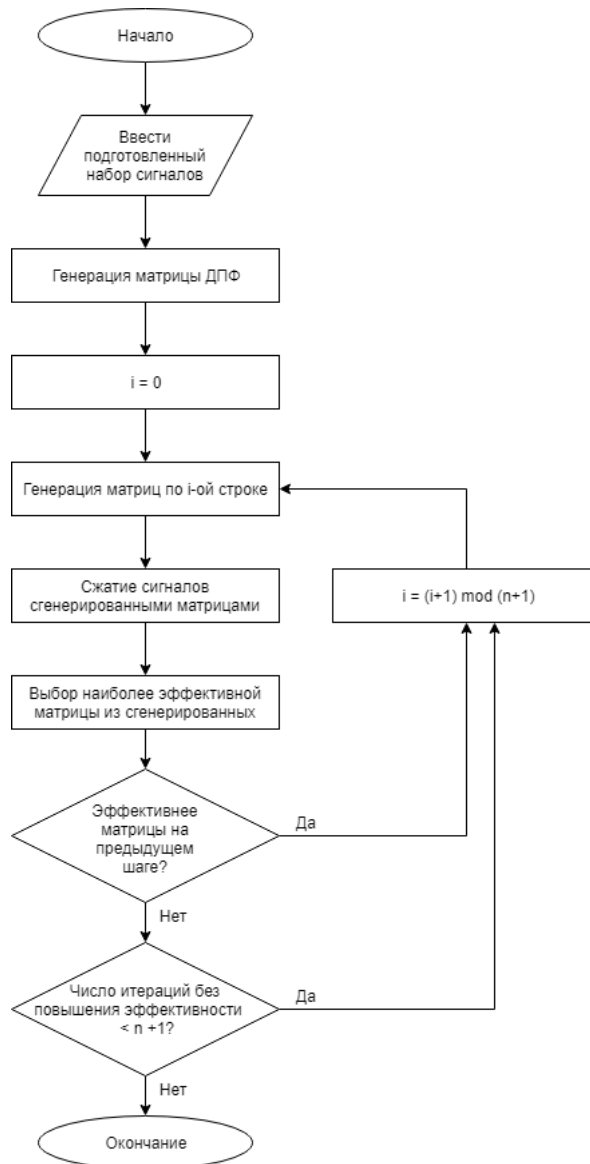


Рис. 1. Блок-схема алгоритма

#### 4.1. Выбор тестовых сигналов и уровня сжатия.

Для решения поставленной задачи исследование проводится на наборе из  $N = 20$  сигналов  $F_p$ . Уровень сжатия выбран равным 80%.

Поиск оптимального алгоритма сжатия имеет большую вычислительную сложность. В связи с этим на этапе предварительных расчетов для уменьшения количества операций рассматриваются сигналы, состоящие из  $n + 1 = 10$  отсчетов.

В исследовании используются сигналы, которые представляют собой набор точек из QRS-комплекса электрокардиограммы (ЭКГ). Данный вид сигнала отражает изменение разницы электрических потенциалов активности сердца во времени и состоит из периодической последовательности кардиоциклов (рис. 2). Одним из элементов такого цикла является QRS-комплекс, который отображает распространение возбуждения по желудочкам. Для проведения исследования были сняты ЭКГ с нескольких людей. Тестовые сигналы сгенерированы путем выбора из различных QRS-комплексов обозначенного ранее количества точек.

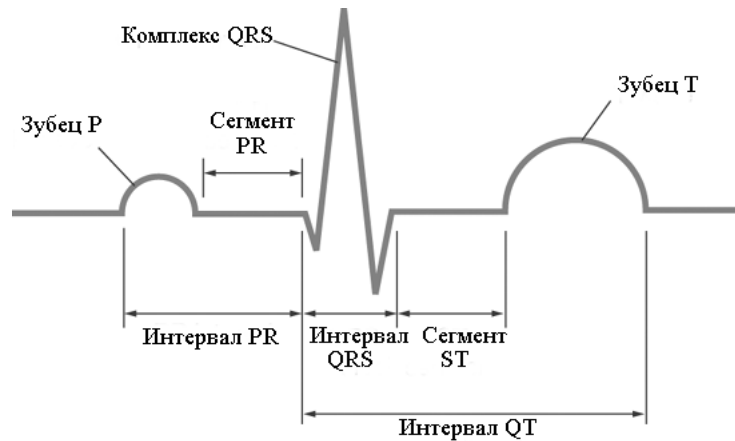


Рис. 2. Кардиоцикл ЭКГ

## 4.2. Генерация матриц преобразования

В качестве начального приближения мы выбираем матрицу дискретного косинусного преобразования Фурье (6), поскольку она дает достаточно хорошие результаты для широкого класса сигналов. Затем с каждым из элементов первой строки выполняем одно из 3-х действий: прибавляем к нему малую величину  $\epsilon > 0$ , вычитаем  $\epsilon$ , либо оставляем элемент неизменным (в наших расчетах  $\epsilon = 0,01$ ). В результате получается набор из  $3^{n+1}$  матриц  $S_q$ , с которыми выполняются расчеты на всех дальнейших этапах.

Для каждой матрицы мы с помощью метода Гаусса находим обратную  $T = S_q^{-1}$ . В случае, если  $S_q$  оказалась вырожденной, либо ее определитель близок к нулю, мы исключаем ее из дальнейшего рассмотрения, поскольку расчеты с участием плохо обусловленных матриц обладают известной вычислительной неустойчивостью [6, 7].

При повторном возвращении на этап генерации матриц преобразования мы производим аналогичные действия со второй строкой, третьей и т. д. Затем весь процесс повторяется итерационно, пока не будут достигнуты условия завершения алгоритма.

## 4.3. Анализ, сжатие и синтез тестовых сигналов.

На данном этапе последовательно перебираются матрицы  $S_q$ . С помощью каждой матрицы производится сжатия всех тестовых сигналов  $F_p, p = 1, 2, \dots, N$ . Обозначим  $G_{qp} = S_q * F_p$ . Далее производится обнуление части элементов векторов  $G_{qp}$  в соответствии с выбранным уровнем сжатия. Новый набор векторов обозначим  $\tilde{G}_{qp}$ . После этого, применяя формулу (2), получаем синтезированные сигналы  $\tilde{F}_{qp} = T_q * \tilde{G}_{qp}$ .

## 4.4. Выбор наиболее эффективного преобразования.

Для каждой матрицы  $S_q$  мы рассчитываем среднюю по всему набору тестовых сигналов среднеквадратичную ошибку  $< \sigma_q >$ :

$$< \sigma_q > = \frac{\sum_{p=1}^N \|F_{qp} - \tilde{F}_{qp}\|}{N} \quad (7)$$

Наиболее эффективным мы считаем преобразование  $S_q$ , которое при заданном уровне сжатия дает наименьшее значение величины  $< \sigma_q >$ . В случае, если  $S_q$  не позволяет достичь отклонения меньшего, чем при использовании ДПФ, либо оно существенно не снизилось после проведения нескольких итераций, то алгоритм завершается. В противном случае мы возвращаемся к этапу генерации матриц преобразования, взяв за основу при дальнейших

манипуляциях со строками найденную матрицу  $S_q$ .

## 5. Заключение

В данной работе нами разработан новый алгоритм поиска оптимального метода сжатия сигналов заданного вида. Для реализации алгоритма используется суперкомпьютер, поскольку его вычислительная сложность достаточно высока. Ключевым достоинством предложенного подхода является, во-первых, универсальность: программа в автоматическом режиме должна осуществлять поиск наиболее эффективной матрицы преобразования. Во-вторых, алгоритм достаточно прост в реализации.

Дальнейшее исследование будет посвящено изучению устойчивости алгоритма: насколько сильно зависит полученная матрица от состава и количества тестовых сигналов. Также планируется провести вычислительные эксперименты в ситуациях, когда вместо ДПФ в качестве начального приближения выбраны какие-либо матрицы, соответствующие различным вейвлет преобразованиям.

## 6. Список литературы

1. Залманзон Л.А. Преобразования Фурье, Волша, Хаара и их применение в управлении, связи и других областях / Л.А. Залманзон. – Москва: Наука, 1989. – 496 с.
2. Bracewell R. The Fourier Transform and its Applications. McGraw-Hill, 2000, 640 p.
3. Addison P. Wavelet Transforms and the ECG: a Review. Physiological Measurement, 2005, № 26(5), pp. R155–R199. DOI: 10.1088/0967-3334/26/5/R01.
4. Короновский А.А. Непрерывный вейвлетный анализ и его приложения / А.А. Короновский, А.Е. Храмов. – Москва: Физматлит, 2003. – 176 с.
5. Власова Е.А. Ряды / Е.А. Власова. – Москва: Изд-во МГТУ им. Н.Э. Баумана, 2006. – 616 с.
6. Кострикин А.И. Линейная алгебра и геометрия / А.И. Кострикин, Ю.И. Манин. – Москва: Наука, 1986. – 304 с.
7. Бахвалов Н.С. Численные методы / Н.С. Бахвалов, Н.П. Жидков, Г.М. Кобельков. – Москва: Наука, 1987. – 598 с.