Поиск оптимального алгоритма сжатия сигнала на системах с общей памятью

А.В. Данилова[1], Н.В. Пауков[1], М.А. Теплякова[1]

[1]Воронежский государственный университет

Аннотация содержит краткое описание статьи и не должна превышать 10 строк. Она оформляется шрифтом размером 10 пт с отступом 15 мм слева и справа, выравниванием по ширине. Сверху и снизу аннотация отделяется от остального текста интервалом в 1 строку размером 16 пт. Настоящий текст оформлен в соответствии со всеми требованиями и может быть использован в качестве шаблона.

Ключевые слова: дискретное косинусное преобразование Фурье, сжатие сигналов, параллельное программирование

1. Введение

Обработка цифровых сигналов, несомненно, является перспективной и крайне важной областью научных исследований. Обработка сигнала позволяет избавиться от постороннего шума, появляющегося в результате неточных измерений, фоновых помех или неисправности аппаратуры, и тем самым получить только полезную информацию из сигнала. Кроме того, обработка сигнала может использоваться для сжатия сигналов, что также важно на практике, так как это позволяет сохранить всю важную информацию, присутствующую в сигнале, сократив при этом объём памяти, используемый для её хранения.

Многие алгоритмы сжатия основаны на использовании дискретного косинусного преобразования Фурье(1). Т.е. для широкого спектра сигналов вышеуказанное преобразование является эффективным. Кроме прочего, данное преообрзование широко используется и в реализации многих алгоритмов сжатия изображения и видео, таких как JPEG, MPEG и прочих.

$$\hat{f}_k = \frac{1}{\sqrt{2n+1}}(f_0 + 2\sum_{m=1}^{n} f_m \cos(\frac{2\pi km}{2n+1})) \tag{1}$$

Матричный вид косинусного преобразования Фурье:

$$\frac{1}{\sqrt{2n+1}} \begin{pmatrix} 1 & 2 & 2 & \dots & 2 \\ 1 & 2\cos\frac{2\pi}{2n+1} & 2\cos\frac{2\pi 2}{2n+1} & \dots & 2\cos\frac{2\pi n}{2n+1} \\ \vdots & \vdots & \ddots & \vdots & \\ 1 & 2\cos\frac{2\pi 2}{2n+1} & 2\cos\frac{2\pi 4}{2n+1} & \dots & 2\cos\frac{2\pi 2n}{2n+1} \end{pmatrix}$$

Но существенным аспектом алгоритмов сжатия сигналов является тот факт, что для каждого типа сигнала оптимальным является лишь свой конкретный метод сжатия. Следовательно, вопрос автоматического отыскания оптимального метода сжатия для конкретного типа сигнала представляет серьёзный практический интерес. Таким образом, целью нашего исследования мы решили поставить создание подобного метода. При наличии значительных вычислительных ресурсов, данную задачу можно было бы решить простым перебором различных матриц преобразования, но в реальных условиях это заняло бы слишком большое количество времени. Помимо того, немаловажно чётко определить то, какие матрицы мы будем считать оптимальными. Таким образом, что касается конкретных задач, вставших перед нами, то стоит отметить следующие:

- Создание эффективного алгоритма для поиска оптимальных матриц преобразования
- Выбор критерия для определения оптимальности матрицы преобразования

Так как поиск оптимальных матриц является ресурсозатратной задачей, то для её реализации было принято решение использовать алгоритмы параллельного программирования для систем с общей памятью. В качестве наиболее доступного программного обеспечения была взята распространённая реализация стандарта MP - OpenMP. Все вычисления были проведены с использованием ресурсов Воронежского суперкомпьютерного центра.

2. Методика

При решении вышеозначенной задачи сжатие сигналов осуществляется следующим образом. Берётся сигнал, представленный в векторном виде, и матрица преобразования соответствующего размера. Матрица преобразования проверяется на вырожденность, путём рассчёта её определителя. В случае, если её определитель равен нулю, дальнейшие действия не производится и берётся другая матрица преобразования. Если же определитель отличен от нуля, то матрица S умножается на сигнал $X(2)$.

$$Y = S * X \tag{2}$$

Далее проводится фильтрация полученного преобразованного сигнала Y и, в зависимости от желаемой степени сжатия, фиксированная часть значений в сигнале, являющихся наименьшими по модулю, зануляется. Тем самым отбрасывается избыточная информация. Полученный после проведения фильтрации сигнал обозначим как $\tilde{Y}$. После этого производится обратное преобразования сигнала $\tilde{Y}$, путём его умножения на матрицу S^{-1}, обратную к первоначально взятой матрице преобразования $S(3)$.

$$\tilde{X} = S^{-1} * \tilde{Y} \tag{3}$$

Чтобы определить степень сжатия, вычислим ошибку между исходным и полученным сигналом(4). Оптимальной будем считать матрицу преобразования, имеющую среднюю ошибку меньшую, чем дискретное косинусное преобразование Фурье, или же близкую к нему.

$$\delta = \frac{||X - \tilde{X}||^2}{n} \tag{4}$$

3. Алгоритм

3.1. Генерация сигналов

Для решения поставленной задачи исследование проводится на наборе из 20 сигналов. Поиск оптимального алгоритма сжатия имеет большую вычислительную сложность. В связи с этим для уменьшения количества операций рассматриваются сигналы, состоящие из 8 точек. В исследовании используются сигналы, которые представляют собой набор точек из QRS-комплекса ЭКГ.

3.2. Генерация матриц преобразования

В рамках исследования в качестве элементов матрицы преобразования T берутся числа 1, 0 и -1. Для выбора оптимальной матрицы следует рассмотреть все матрицы T, которые можно составить из выбранных элементов. Однако решение данной задачи для сигналов, состоящих из незначительного числа точек, требует большого количества вычислительных ресурсов. Для уменьшения объема вычислений предлагается проводить классификацию генерируемых матриц.

3.2.1. Первый этап

На начальном этапе матрицы выделяются в класс по количеству входящих в их состав элементов. Обозначим I_0 – класс матриц, в которых не встречается число 1, I_1 - класс матриц, в которых число 1 встречается один раз. Поскольку рассматриваются сигналы длиной 8 точек, последним классом будет I_{64}, в который входит матрица, состоящая только из единиц. Далее для каждого из выделенных классов необходимо сгенерировать несколько соответствующих им матриц, провести преобразование сигналов и на основании полученных результатов выявить наиболее эффективный из классов.

Следующим действием класс матриц, который оказался наиболее оптимальным, необходимо разделить на подклассы по количеству числа -1. Пусть n – число единиц в рассматриваемых матрицах, тогда новые классы разбиения обозначим как NI_0^n, NI_1^n, ..., NI_{64}^n. Далее также генерируются несколько матриц для каждого класса, и проводится исследование, в результате которого будет определено оптимальное количество числа -1 (обозначим его m). Таким образом, на данном этапе будет выделен класс матриц NI_m^n, который наилучшим образом подходит для решения поставленной задачи. Для найденного класса известен точный состав матриц: количество числа 1 равно n, количество числа -1 составляет m, количество числа 0 обозначим k, которое определяется по формуле (5).

$$k = 64 - n - m \tag{5}$$

3.2.2. Второй этап

На данном этапе матрицы классифицируются с точки зрения распределения элементов по строкам. Пусть исследование начинается с числа 1. Рассматривается первая строка. Обозначим R_0^1 – класс матриц, в первой строке которых число 1 не встречается, R_1^1 – класс матриц, в первой строке которых число 1 встречается один раз. Последним классом будет R_8^2, ему соответствуют матрицы, первая строка которых состоит только из единиц. Для каждого класса генерируется несколько матриц таким образом, чтобы в первой строке было соответствующее рассматриваемому классу количество элементов, равных 1, а в остальных строках они были случайно распределены. Затем выделяется тот класс, который наиболее оптимально решает поставленную задачу. Далее рассматривается следующая строка, и для нее аналогично выделяются классы R_0^2, R_1^2, ..., R_8^2. При генерации матриц первая строка заполняется согласно требованиям класса, который был выбран на предыдущем шаге, вторая строка должна соответствовать текущему классу, остальные заполняются случайным образом, при этом общее количество числа 1 должно быть равно n. Такая классификация проводится для всех строк матрицы.

Затем проводится классификация по распределению числа -1 по аналогии с тем, как проводилось для 1. В результате для всех строк матрицы будет известно количество каждого элемента, которые ее составляют (обозначим n_i, m_i, k_i для 1, -1, 0 соответственно, где $i = 1, 2, \ldots, 8$ – номера строк).

3.2.3. Третий этап

Это последний этап классификации, на котором рассматривается расположение элементов в строке. Рассматривается первая строка. С помощью перестановки элементов формируются различные комбинации их расположения в строке, каждый из которых образует свой класс матриц. Среди них проводится поиск наилучшей. Аналогичные действия производятся для каждой строки.

В результате третьего этапа классификации будет известен точный вид матрицы, наилучшим образом решающей поставленную задачу.