

УДК 621.391

А.В. Новиков

Метод сжатия цифровых данных широкополосного радиолокатора

Предложен метод сжатия цифровых радиолокационных данных с коэффициентом сжатия $3 \pm 0,04$ и возможностью восстановления радиолокационной отражаемости с ошибкой не более 1%. Показана возможность выполнения соответствующего алгоритма в реальном времени работы радара, т.е. когда соответствующие временные затраты соизмеримы с затратами на базовые операции радара.

Ключевые слова: широкополосная радиолокация, алгоритм Лемпела–Зива–Велча, хэш-таблица.

Постановка задачи. В начале XXI в. в связи с достаточным развитием цифровой техники (особенно её метрологии) получили широкое распространение радиолокаторы высокого разрешения (единицы метров), использующие для зондирования сложные сигналы с линейной частотной, фазо-кодовой и тому подобными модуляциями. Зачастую такие локаторы объединяют в сеть для более качественного обзора территорий с затеняющими объектами. Сферы применения таких радаров – это, например, речные пароходства, морские суда, территории с охраной периметра, площадки с большим количеством подвижных объектов (железнодорожные станции).

Одной из основных проблем является необходимость передачи в реальном времени большого потока цифровой информации (десятки Мбит/с) по существующим каналам связи. Примером здесь может послужить потребность охвата территории крупной железнодорожной станции, при этом данные со всех радаров должны стекаться в одном месте для показа радиолокационного изображения на индикаторах (чаще всего кругового обзора).

Таким образом, решаемая здесь задача – предложить метод и с его помощью показать возможность и степень сокращения объема передаваемых радиолокационных данных в реальном времени и с допустимой ошибкой восстановления исходных данных.

Принципы получения данных с радара. Рассматривается радиолокационная станция (РЛС) «Река» производства ЗАО НПФ «Микран».

Данная РЛС с определённой частотой повторения (~280 Гц) посылает в окружающее пространство зондирующий сигнал и одновременно принимает отражённый, несущий информацию об объектах местности. Принимаемые данные можно разделить на блоки, каждый из которых содержит информацию об объектах местности, расположенных на линии визирования антенны радара. Такой блок в данной работе называется дальностной строкой.

Отсчёты амплитуд в дальностной строке идут с равномерным шагом по дальности, зависящим от текущей разрешающей способности, которая, в свою очередь, зависит от ширины спектра излучаемого сигнала. Для РЛС «Река» количество отсчётов по дальности фиксировано, поэтому объём цифровой информации не зависит от текущего разрешения по дальности и определяется форматом представления амплитуды.

Скорость передачи данных РЛС «Река» в режиме без потерь составляет 37,5 Мбит/с, что составляет около 40% от пропускной способности всё ещё широко распространённой сети «Ethernet 100 Мбит». Разрешение рассматриваемого радара не является предельным. Технически возможно создание радара с более высоким разрешением, например, в 2,5 раза большим, при этом объём данных увеличится во столько же раз и сеть «Ethernet 100 Мбит» уже не справится. Данное ограничение призвано уменьшить методы сжатия цифровых данных.

В настоящее время, как правило, методы создаются на основании подходящей комбинации множества уже известных методов. В данной работе этот принцип активно используется. Разрабатываемый здесь метод направлен на решение определённой задачи.

Физические возможности сжатия радиолокационной информации. Можно выделить следующие физические возможности для сжатия:

- различимость глазом человека конечного диапазона яркостей (в цифровом телевидении выделено 220 уровней для сигнала яркости);

- ненулевая дисперсия амплитуды отражённого от некоторого объекта сигнала.

Оценка дисперсии амплитуды по четырём-пяти оборотам антенны составила 12% относительно самой амплитуды. Данный фактор является доминирующим, так как выделение 220 уровней яркости допускает ошибку 0,5% (один дискрет по отношению к максимуму).

Таким образом, случайные изменения яркости одного и того же объекта оказываются больше порога различимости человеческого глаза.

Технические возможности сжатия радиолокационной информации. Информация об амплитуде, в конечном итоге определяющей яркость цели, в РЛС «Река» хранится в виде числа с плавающей точкой одинарной точности размером 4 байта (далее обозначается как «float»). Необходимость выбранного формата продиктована существующим стандартом на вещественные числа [1]. Однако, исходя из вышеприведённых оценок дисперсии, точность данного формата весьма избыточна (7–8 знаков после запятой).

Первая возможность для сжатия – отбросить часть мантиссы числа float, а именно: оставить два старших байта из четырёх

$$\underbrace{se_1e_2e_3e_4e_5e_6e_7}_{4\text{-й байт}} \quad \underbrace{e_8m_1m_2m_3m_4m_5m_6m_7}_{3\text{-й байт}} \quad \underbrace{xxxxxxx}_{2\text{-й байт}} \quad \underbrace{xxxxxxxx}_{1\text{-й байт}}, \quad (1)$$

где s – знаковый бит (всегда нулевой, так как амплитуды положительны); e_i – биты экспоненты; m_i – биты мантиссы. Далее для упрощения четвёртый байт (1) будет называться «экспонента», а третий – «мантисса».

Ошибка округления (1) вычисляется следующим образом.

Примем за исходное число (за эталон) положительную единицу, которая в формате (1) запишется как

$$0\ 1111111\ 0000000\ 000000000000000000000000 \quad (2)$$

Максимальная ошибка при усечении двух байт будет при заполнении шестнадцати крайних бит единицами (выделено жирным)

$$0\ 1111111\ 0000000\ 1111111111111111 \quad (3)$$

Делая по формуле из [1] пересчёт (2) и (3) в десятичные числа, получим величину ошибки

$$\left(1 - \left(1 + 65535/2^{23}\right)\right) \approx 0,0078. \quad (4)$$

Восстановление сжатого числа до исходного (распаковка) происходит вставкой двух нулевых недостающих байт.

Таким образом, идея отброса двух младших байт даёт сокращение объёма передаваемой информации в два раза при амплитудных ошибках не более 1%. Очевидно, что затраты на сжатие-распаковку здесь ничтожны, так как все операции выполняются с данными, **кратными байту**. Данный способ был опробован и успешно реализован в программном обеспечении РЛС «Река».

Вторая возможность для сжатия возникает при попытке ответа на вопрос о том, можно ли набор двухбайтовых чисел сжать ещё?

Если первая идея привела к алгоритму сжатия с потерями, то теперь потери недопустимы, хотя бы потому, что в цифровой вычислительной технике информацию принципиально передают кратко байту. Остаются алгоритмы сжатия без потерь, которые, естественно, уступают по вычислительным затратам. Здесь необходимо постараться выбрать такой способ сжатия, который имеет допустимые вычислительные затраты при как можно большем коэффициенте сжатия.

Идея сжатия без потерь, как известно, базируется на статистике символов, несущих некоторую информацию. Как было сказано, кратность байту – базовое ограничение вычислительной техники, поэтому из формулы (1) следует, что можно набрать статистику для символов, соответствующих четвёртому и третьему байтам. Результаты статистического анализа (рис. 1) говорят о потенциальной возможности сжатия без потерь только потока экспонент (четвёртый байт float).

Для порядка 10000 линий дальности была оценена энтропия (рис. 2) для потоков экспонент и мантисс. Анализ показывает, что потенциальный коэффициент сжатия потока экспонент составляет $8/1,75 \approx 4,57$ раза. Энтропия потока мантисс практически достигает максимального значения (8 бит/символ), поэтому сжимать этот поток не имеет смысла.

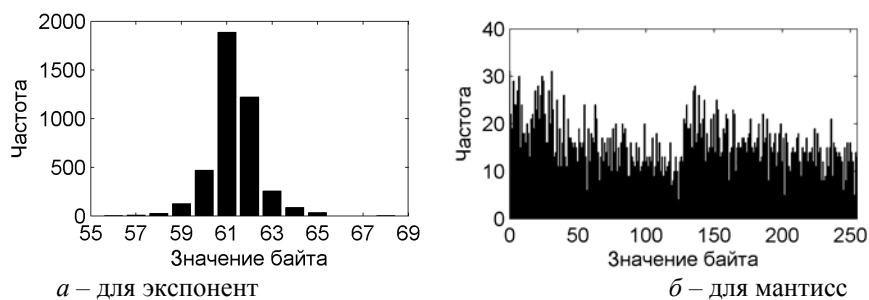


Рис. 1. Примеры частот повторяемости байтовых символов в дальностной строке РЛС «Река»

Таким образом, в блоке сжатия без потерь намечается блок, выделяющийся из линии дальности **поток экспонент**, который далее поступает в сжимающий кодер, и **поток мантисс**, передаваемый в канал связи напрямую. Операции разделения и сбора байт также не вносят значительных вычислительных расходов, так как размер символов кратен байту.

Можно заметить, что одной из базовых идей при сжатии является известная идея декомпозиции (раздельного анализа частей какого-либо объекта, в данном случае – числа с плавающей точкой).

Самое простое при выборе методов сжатия без потерь – это кодек длин повторов, Run Length Encoding, RLE [2]. Однако реализация RLE показала небольшой коэффициент сжатия на радиолокационных данных РЛС «Река» (1,3–1,4 в сравнении с потенциально возможным коэффициентом 4,57). Поэтому следующим шагом был анализ идеи сжатия Лемпела–Зива [2].

Существует множество вариантов сжимающих кодеков, основанных на этой идее.

Сначала автором был рассмотрен метод Lempel–Ziv 78 (LZ78) [2], заключающийся в динамическом составлении таблицы-словаря и передаче в канал пар (номер, символ). Затем, как более практичный вариант, требующий передачи только номера, был проанализирован метод Lempel–Ziv–Welch (LZW) [2], используемый, например, при кодировании изображений в формат Graphics Interchange Format (GIF), опционально в Tagged Image File Format (TIFF), а также при создании документов Portable Document Format (PDF) и PostScript. Два этих метода защищены патентами разных стран (США, Канада, Япония), но к настоящему времени их сроки истекли.

Средние коэффициенты сжатия для методов LZ78 и LZW получились равными 2,4 и 3,2 соответственно. Очевидно преимущество LZW (для этой цели, а также для упрощения реализации и был модифицирован метод LZ78).

Что касается вычислительных затрат, то прямая реализация методов словарного сжатия LZ78 и LZW, когда поиск по таблице-словарю происходит линейно (сверху вниз, снизу вверх – без разницы), показала недостаточные скорости компрессии и декомпрессии для работы в реальном времени радара (получилось в 15–20 раз медленнее базовой операции радара). Поэтому следующим шагом к достижению цели была оптимизация операции поиска по таблице. Выбор, естественно, пал на хэш-таблицы [3], дающие константное время поиска любого элемента таблицы.

В качестве хэш-функции для вычисления индекса была выбрана [3]

$$i(k) = \lfloor N \cdot \{\varphi \cdot k\} \rfloor, \tag{5}$$

где k – ключ для поиска, формируемый по исходным данным; N – количество элементов в хэш-таблице; φ – золотое сечение 0,618...; $\{\cdot\}$ – функция взятия дробной части; $\lfloor \cdot \rfloor$ – функция взятия целой части «пол» (англ. «floor»).

Ключ k для поиска формируется по правилу

$$k = (s \ll n) \vee p, \tag{6}$$

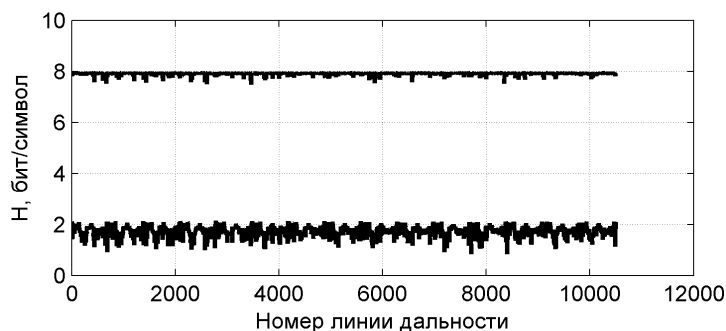


Рис. 2. Пример зависимостей энтропии байтовых символов для амплитудных данных РЛС «Река». Верхняя кривая – для мантисс, нижняя – для экспонент

где s – заносимый в таблицу байтовый символ; n – количество бит, отводимых под номер, непосредственно выдаваемый сжимающим кодером; p – номер из строки таблицы, где был найден последний отрезок; \llcorner – операция побитового сдвига влево; \vee – операция побитового ИЛИ.

Проверка хэш-функции (5) на реальных данных радара позволила оценить вероятность коллизии в 0,09, при этом оказались задействованы не более трёх-четырёх уровней хэш-таблицы. Другими словами, вероятность того, что элемент будет найден в таблице за одну попытку, составила 0,91, при этом изредка возможны две попытки, ещё реже – три и четыре.

Тестовая реализация метода LZW с использованием (5) показала возможность сжатия данных в реальном времени работы радара: в 1,5 раза медленнее базовой операции радара.

Объём памяти, требуемый для хранения хэш-таблицы, зависит от битности кода n в (6). В качестве оптимальной (по максимуму среднего коэффициента сжатия) выбрана длина кода $n = 10$, при этом хэш-таблица состоит из 2048 элементов-троек типа $\langle p, s, c \rangle$, где c – номер текущей строки.

С учётом того, что сначала данные сжимаются (с потерями) в два раза, а затем половина от сжатых данных жмётся кодером LZW в $3,2 \pm 0,19$ раза, общий коэффициент сжатия составит

$$K_T = \frac{4 \cdot K_{LZW}}{1 + K_{LZW}} = \frac{4 \cdot 3,2}{1 + 3,2} \approx 3,0, \quad \Delta K_T \approx \frac{4}{(1 + K_{LZW})^2} \cdot \Delta K_{LZW} = \frac{4}{(1 + 3,2)^2} \cdot 0,19 \approx 0,04.$$

Разделение данных на экспоненты и мантиссы дополнительно позволило уменьшить дисперсию коэффициента сжатия.

В программном обеспечении РЛС «Река» метод сжатия без потерь пока не реализован.

Пример сжатия реальных радиолокационных данных

В качестве примера рассмотрим 16 двухбайтовых отсчётов (сжатие с потерями уже произошло)

68 170 63 148 63 195 63 186 63 5 62 105 63 108 61 128 62 24 62 93 61 188
62 45 59 225 61 68 61 28 60 130.

Блок, разделяющий мантиссы и экспоненты, выделит поток экспонент

68 63 63 63 63 62 63 61 62 62 61 62 59 61 61 60,

который поступает в кодер LZW. На рассматриваемом блоке данных сжатия может и не наблюдаться, так как блок короткий, однако это не мешает понять сам алгоритм (таблица).

Пример кодирования потока экспонент кодом LZW

Отрезок	Номер (код)	Отрезок	Номер (код)
0	0	63 62	261
1	1	62 63	262
2	2	63 61	263
...	...	61 62	264
255	255	62 62	265
Clear	256	62 61	266
End	257	61 62 59	267
68 63	258	59 61	268
63 63	259	61 61	269
63 63 63	260	61 60	270
Вход кодера			
Выход кодера			
256, 68, 63, 259, 63, 62, 63, 61, 62, 62, 264, 59, 61, 61, 60, 257			

В начале кодирования первые 256 строк таблицы всегда заполняются всевозможными байтовыми символами с кодами (0–255). Далее идут два служебных символа: символ очистки таблицы **Clear** с кодом 256 и символ-признак конца архива **End** (код 257). Затем таблица динамически заполняется наборами чисел по следующему алгоритму.

Поступающий символ ищется в таблице (в рассматриваемом случае это первый символ (68)), и если он там есть, считывается следующий (63). В таблице ищется отрезок (68, 63), которого в данном случае там нет. Отрезок заносится в таблицу и ему присваивается следующий номер по поряд-

ку, т.е. 258. Далее считывается символ (63), который, естественно, всегда есть в таблице, и считывается следующий символ (63). Отрезка (63, 63) там нет, поэтому ему присваивается номер 259. Далее идёт похожая ситуация, т.е. отрезок (63) – есть, (63, 63) – есть, а вот отрезка (63, 63, 63) – нет в таблице, поэтому он туда заносится со следующим свободным номером 260. Процесс продолжается до тех пор, пока не закончатся данные или пока не переполнится таблица. В случае переполнения на выход кодера выставляется символ очистки и таблица сбрасывается.

По сути, в таблице-словаре необходимо хранить лишь тройки чисел:

- номер последнего найденного в таблице отрезка. Например, для (63, 63) это 259;
- последний символ в отрезке, которого нет в таблице (это **63** для отрезка (63, 63, **63**));
- номер текущего отрезка (260).

В этом случае структура таблицы будет значительно проще и её объём будет заранее определён.

Для рассмотренного примера коэффициент сжатия получился равным единице (16 символов на входе, 16 на выходе). Это связано с небольшой длиной блока. В РЛС «Река» дальностная строка состоит из 4096 отсчётов и приведённые в заключении результаты предполагают этот факт.

Заключение. Показано, что в реальном времени работы РЛС «Река» возможно обеспечить сжатие радиолокационной информации с ошибкой восстановления радиолокационной отражаемости не более 1% и коэффициентом сжатия $3 \pm 0,04$, что даёт средний цифровой поток $12,5 \pm 0,17$ Мбит/с против исходного 37,5 Мбит/с ($\sim 1,17 \cdot 10^6$ чисел с плавающей точкой в секунду, ~ 11 бит/отсчёт). При этом дополнительные вычислительные затраты на сжатие составляют 1,5 от затрат на базовую операцию радара.

Предварительный анализ известного [2] метода разностного сжатия радиолокационных данных, когда используется корреляция дальностных строк, показал неудовлетворительный результат. Попытки кодирования с линейным предсказанием дали результаты, которые не лучше результатов без предсказания (анализ проводился при максимальной скорости вращения антенны – 24 об/мин).

В данной работе проведён анализ возможностей сжатия цифровых данных для конкретного радара. Однако, основываясь на общих свойствах чисел с плавающей точкой и примерно одинаковом динамическом диапазоне радаров, можно с уверенностью предположить возможность использования предложенного метода в других широкополосных радарах.

Литература

1. IEEE 754 – стандарт двоичной арифметики с плавающей точкой [Электронный ресурс]. – Режим доступа: <http://www.softelectro.ru/ieee754.html>, свободный (дата обращения: 19.06.2013).
2. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео / Д. Ватолин, А. Ратушняк, М. Смирнов, В. Юкин. – М.: ДИАЛОГ-МИФИ, 2003. – 384 с.
3. Алгоритмы: построение и анализ = Introduction to Algorithms / Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Штайн. – 2-е изд. – М.: Вильямс, 2005. – 1296 с.

Новиков Анатолий Викторович

Канд. техн. наук, ассистент каф. радиотехнических систем ТУСУРа, инженер ЗАО НПФ «Микран»

Тел.: 8-952-155-96-99

Эл. почта: novikov@micran.ru, naw1983@yandex.ru

Novikov A.V.

Broadband Radar Data Compression Method

In the paper we offer a real-time algorithm for lossy radar data compression with a compression ratio of 3 ± 0.04 . The recovery error of the radar reflectivity is less than 1%. The time required for the algorithm is comparable to the costs of the basic operation of the radar.

Keywords: broadband radar, Lempel–Ziv–Welch algorithm, hash table.