

## Искажения и потери: провалы в работе с данными сейсморазведки Corruption and loss: pitfalls in seismic data management

**Можно ли доверять данным? Элеонора Джек (Eleanor Jack), старший геофизик компании Landmark Graphics, приводит ряд поучительных соображений о том, что может произойти с записанными данными и что делать для улучшения управления данными и их защиты**

### Типы повреждений

Обычно считается (во всяком случае, так считает владелец данных), что при копировании данных с носителя на носитель все данные с оригинала благополучно переносятся в место назначения. Руководства по управлению данными обычно поддерживают эту мысль, демонстрируя блок-диаграммы переноса данных, где квадратики и стрелочки обозначают перенос данных, который, как предполагается, происходит целиком и без проблем. Дело, к сожалению, не всегда обстоит именно так, чему и посвящена эта статья. Данные могут быть утрачены или повреждены, и это повреждение ждет, затаившись, когда эти данные понадобятся.

Перезапись и копирование данных обычно производятся специализированными компаниями. Такая компания сможет обеспечить наиболее эффективную передачу данных с использованием проверенного программного обеспечения для перевода данных из различных форматов, применяемых в сборе и обработке данных в стандартный формат обмена данными SEG-Y. Все приличные компании производят также ту или иную проверку качества. Тем не менее, ограничения по расходам и неполное понимание природы данных и их форматов ведет к тому, что существенная часть данных сохраняется со скрытыми ошибками или потерями. Далее разобран ряд примеров ошибок, прошедших через первичную проверку качества.

Причины повреждений делятся на два основных типа: технические (не зависящие от человека – плохая лента, сбой аппаратуры), и явно связанные с «человеческим фактором» – ошибками человека. В этой отрасли, как и везде, работает закон Мэрфи: если ошибку можно совершить, кто-нибудь ее совершит.

Наиболее часто встречается повреждение данных из-за неверного определения размеров блока. На магнитной ленте в файл формат SEG-Y трассы физически разделены промежутками, которые дают любой прикладной программе четкое указание, где кончается одна трасса и начинается другая. При записи на диск, однако, таких промежутков нет, и единственным средством понять, где конец трассы, является указание количества отсчетов на трассу в заголовке файла. Таким образом, необходимо, чтобы эта величина точно соответствовала длине трассы.

По ряду причин, однако, расхождение данных в заголовке трассы с ее фактической длиной встречается пугающе часто. Наиболее безопасный подход к чтению данных в формате SEG-Y состоит в использовании защищенных (инкапсулированных) форматов, например открытого формата Tare Image Format (TIF) с явной цифровой маркировкой промежутка между трассами, чтобы данные можно было проверить и устранить любые несоответствия в длинах блоков до снятия инкапсуляции.

Несоблюдение этой процедуры приведет к катастрофе.

Неустраненное несоответствие в длине блока в файле приведет, как минимум, к потере всех данных, следующих после ошибки, а в худшем – к невозможности прочесть весь файл. Пример на рис. 1 представляет полевую запись, но эта проблема может возникнуть (и возникает) и в данных после суммирования по ОГТ, а в случае 3D съемки такая ошибка может привести к утрате всех данных.

### Причины несоответствия в длине блоков

Несоответствие в длине блока может возникнуть на любом этапе передачи данных и по многим причинам. Обычный сбой при чтении ленты, при котором, скажем, трасса не дочиталась до конца, приводит к тому, что информация заголовка не соответствует фактическому объему данных. Последствия такой ошибки при копировании будут пагубными: если ее не исправить, файл будет потерян.

При демультимплексировании такая ошибка либо исправляется, либо нет; это зависит от того, как данная программа управляется с ошибками в полевых записях. Если в заголовке трассы указан фактический объем данных, все будет хорошо. Если же в заголовке трассы указано количество прочитанных (а не записанных) байтов, возникает угроза возникновения несоответствия длины.

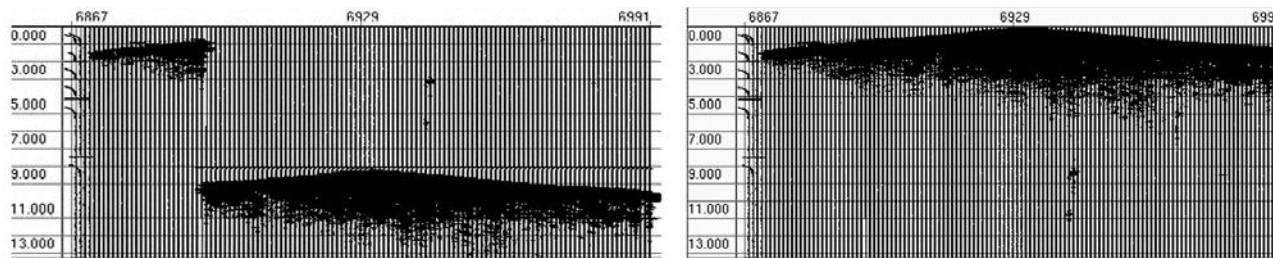


Рис. 1. Полевая запись, в которой длина одной из трасс больше объявленной, сохраненная на диск без исправления (слева) и исправленный вариант (справа). Исправление проводилось вручную путем поиска заголовка следующей трассы в шестнадцатеричном массиве с последующим удалением ненужного куска.

Ошибки в длине блока могут возникать также при повреждении данных в компьютере (например в трассе появляется лишний кусок, который делает ее длиннее, чем нужно или заголовок трассы повреждается так, что искажается значение длины трассы). Появляются такие ошибки и при обработке, поскольку в разных компаниях разные требования и проводятся разные процедуры. По-разному, например, работают с пустыми трассами: иногда в них оставляют только заголовок, иногда они имеют полную длину, но в заголовке указана нулевая длина.

Вообще говоря, раньше фирмы-обработчики расходились во мнениях, где и как указывать в файле длину трассы (формат SEG-Y, к сожалению, позволяет делать это в нескольких местах), у одного из них, в частности, длина трасс после суммирования по ОГТ была неодинаковой. Фактически каждая фирма-обработчик выдавала данные, которые могли читать его программы, но часто ничьи больше.

### Исправление данных

Если данные предназначены для хранения на диске или передачи в реальном времени, очень важно исправить все отклонения от принятого стандарта. Это значит, что все трассы должны быть одинаковой длины, и все сведения о длине трассы, где бы они ни встречались, должны быть верными.

Естественно, каждый, кто занимается исправлением данных, должен точно знать, какая именно из множества возможных причин привела к данной ошибке. Но в любом случае исправление данных следует проводить на этапе чтения с ленты или, по крайней мере, до снятия инкапсуляции, иначе, как в примере на рис. 1, восстановление становится гораздо более затруднительным. Оставлять это на потом нельзя, но большинство, тем не менее, именно так и поступают.

### Потеря данных

Не хватит места, чтобы перечислить 1001 способ потерять данные, но некоторые типичные примеры, особенно близкие всем, могут стать предостережением.

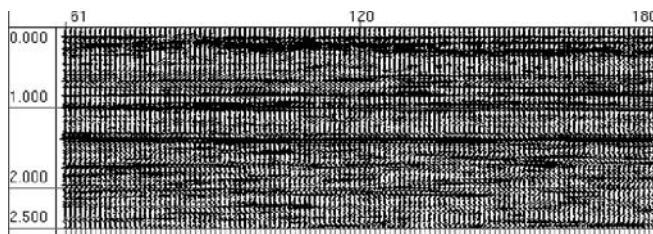


Рис.2. Разрез по данным 3D съемки, обрезанный при перезаписи по времени 2.5 с

Пример на рис. 2 взят из данных 3D съемки в северном море.

При первой попытке сохранить эти данные не было причин думать, что что-то не так, кроме того факта, что все трассы заканчивались на времени 2.5 сек, что, в общем, мало.

Случайно через четыре года эти данные перезаписывали вновь, для другого заказчика, но на этот раз длина записи оказалась 4 сек, и, кроме того, данные, хотя и остались в формате SEG-Y, записались как двухбайтовые числа с фиксированной точкой, а не, как принято, в виде четырехбайтовых чисел с плавающей точкой. Стала ясна возможная причина потери данных: при записи данные были преобразованы к четырехбайтовому виду, но не было учтено что, запись стала вдвое длиннее. Поскольку автор только у себя на компьютере сталкивался с этим явлением четыре раза, то, надо думать, случается такое довольно часто, и многие компании пострадали от этого

### Повреждение из-за формата

Проблема, показанная на рис.3, также возникла из-за неточностей в формате, хотя в этом случае с первого взгляда ясно, что произошло что-то серьезное. Изначально данные были записаны на ленту, причем по ошибке было указано, что данные представлены в виде двухбайтовых чисел с фиксированной точкой. При перезаписи их преобразовали в

четырехбайтовые числа с плавающей точкой, не понимая, что они и так уже четырехбайтовые, и (надемся) не видя разреза. Каждая половина слова была, таким образом, преобразована в четырехбайтовое значение и данные были успешно зашифрованы.

Это пример тем более удручает, что эту ошибку совершали на протяжении двух лет в ходе четырех проектов, а найти ее можно было мгновенно, просто посмотрев на разрез.

### Демультимплексирование

Рассмотренные проблемы обнаруживались в обработанных данных формата SEG-Y. Но в любом случае хранятся по большей части полевые данные, а полевые данные перед записью следует демультимплексировать и преобразовать к формату SEG-Y.

Демультимплексирование полевых данных порождает гораздо больше ошибок, чем простое копирование файлов формата SEG-Y. Данные изначально более сложны и проверка качества не сводится к сравнению входа с выходом, поскольку сами данные меняются. Многие компании используют для проверки сравнение соседних трасс, но так можно пропустить много ошибок. Проблема в том, что данные могут быть весьма похожи на сейсмические, но совсем не являться таковыми.

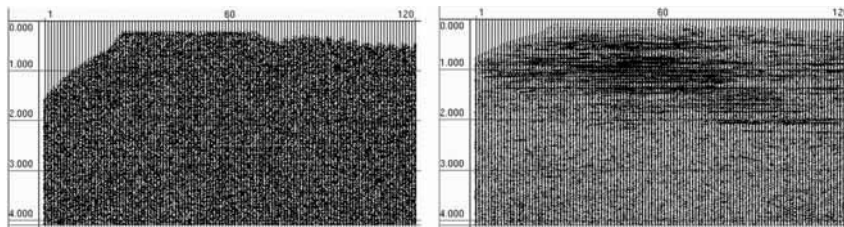


Рис. 3. Данные, зашифрованные при переформатировании с неверным указанием формата (слева). Для восстановления пришлось вернуться к исходным данным и указать верный код формата данных, соответствующий четырехбайтовому числу с плавающей точкой, как и было на самом деле (справа).



## Параметры

### демультиплексирования

Более того, при демультиплексировании пользователь должен сам задавать больше параметров, что повышает вероятность ошибки. Для демультиплексирования нужно задавать длину записи, код формата SEG-Y для вывода, интервал между отсчетами, число трасс на расстановку и формат полевых данных. Может показаться, что все эти параметры критичны и любая ошибка в них сорвет весь процесс. Многие программы демультиплексирования, однако, достаточно устойчивы и доблестно пытаются дать какой-то результат, даже если им скормили ложную информацию.

### Неверное число трасс на расстановку

На рис. 4 показано, что может случиться, если неверно указать число трасс на расстановку при демультиплексировании. С первых 31 трассы первые отсчеты записаны в выходной файл верно, но вторые отсчеты этих трасс записаны на место первых отсчетов трасс с 32 по 61. Третьи отсчеты трасс 1-31 записаны на место вторых и так далее, а потом данные кончились, и половина каждой трассы осталась пустой. В результате получился файл, в котором на расстановку приходится 62 трассы, а каждая пара трасс составлена последовательных отсчетов исходной записи, причем ни один отсчет, кроме первых 31 не находится на своем времени.

Поскольку сами отсчеты переданы верно, и потери данных как таковой не произошло, теоретически возможно восстановить правильные трассы, но оказалось проще и надежнее повторить демультиплексирование.

Хотя в данном случае исполнитель пропустил эту ошибку, ее сравнительно легко выявить при проверке качества. Число трасс на расстановку отличалось от остальных данных, в середине расстановки возникали дополнительные трассы, и вторая половина всех трасс была пустой: все это должно вызвать подозрения.

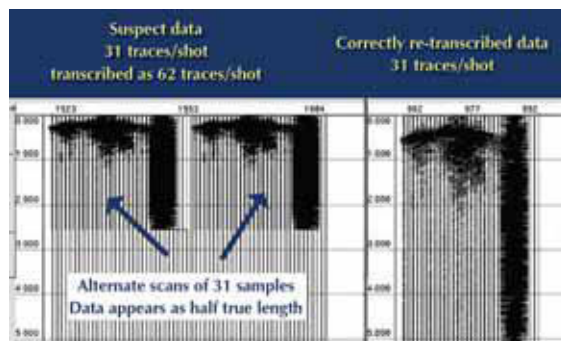


Рис. 4. Неверно перезаписанные данные (слева) и данные после верной повторной записи (справа)

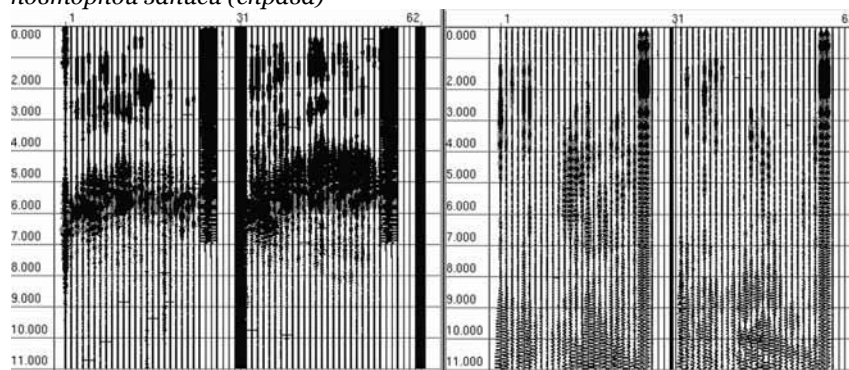


Рис. 5. Данные съемки с источником «вибросейс», перезаписанные с верно указанным интервалом между отсчетами 4 мс (справа) и те же данные с интервалом между отсчетами 2 мс (слева)

Не все программы демультиплексирования дадут в такой ситуации одинаковый результат. Лучше было бы, чтобы данные передавались верно, а пустые трассы шли бы на выход целиком. Опыт автора показывает, что, к сожалению, что чаще случается так, как в рассмотренном примере.

### Неверный интервал между отсчетами

Ошибки, связанные с интервалом между отсчетами, не так просто зафиксировать, как может показаться, особенно, если, как на рис. 5, в качестве источника применяется вибросейс. Для начала, на этом профиле интервал между отсчетами отличается от других, снятых в ходе работ, поэтому при сравнении с другими разрезами это выглядит неправильным. Еще одна улика – свип-сигнал на 25 канале. На других профилях весь свип-сигнал занимает 7 сек, а в проблемном файле он идет до конца записи

Заметим, что времена взяты не из файла, как в формате SEG-Y, а рассчитаны программой визуализации по другим

параметрам, в том числе - интервалу между отсчетами, и, если он указан неверно, время также будет неверно. После исправления интервала между отсчетами (рис. 6) данные стали выглядеть также, как на других профилях, но 5.5 сек записи были утрачены.

В случае противоположной ошибки, если бы был указан интервал 2 мс при фактическом интервале 4 мс, вторая половина каждой трассы была бы пустой, и исправление можно было бы провести без потери данных. Но именно такая ошибка, к сожалению, скорее исключение, чем правило.

### Формат полевой записи

Невероятно, но факт: некоторые полевые форматы можно прочесть как другие форматы, и получить результат, похожий на осмысленный. На рис 7, 8 показано, что получилось из файла в формате Sercel SN348, прочитанного как файл в стандартном формате SEG-Y. Форматы одинаковы, но в Sercel SN348 в заголовках записи нет информации по усилению на каждом канале. Таким образом, к каждой трассе применено неверное значение

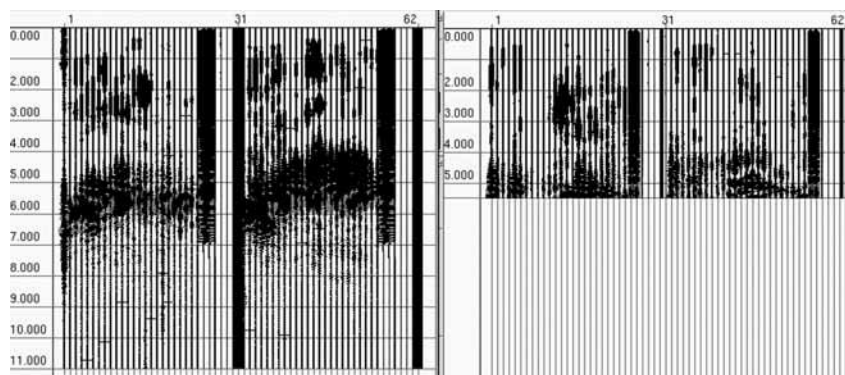


Рис. 6. Сравнение данных с исправленным на 2 мс интервалом между отсчетами (справа) с другими верными данными по тому же участку (слева). 5131 5193 5255

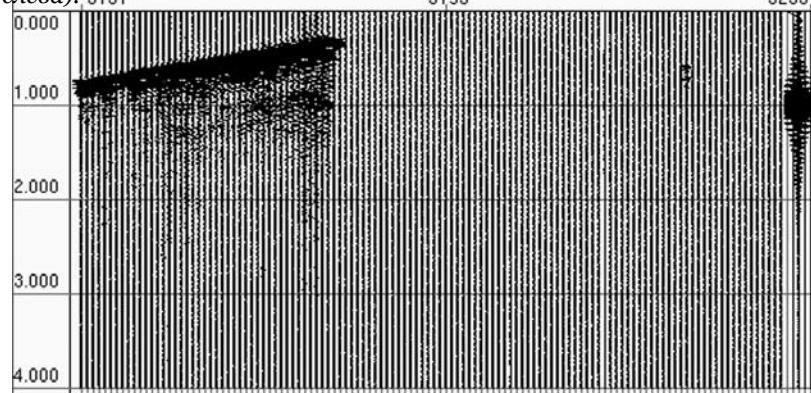


Рис. 7. Файл формат Sercel SN348, прочитанный как файл стандартного формата SEG-B, с использованием первой трассы для определения уровня записи

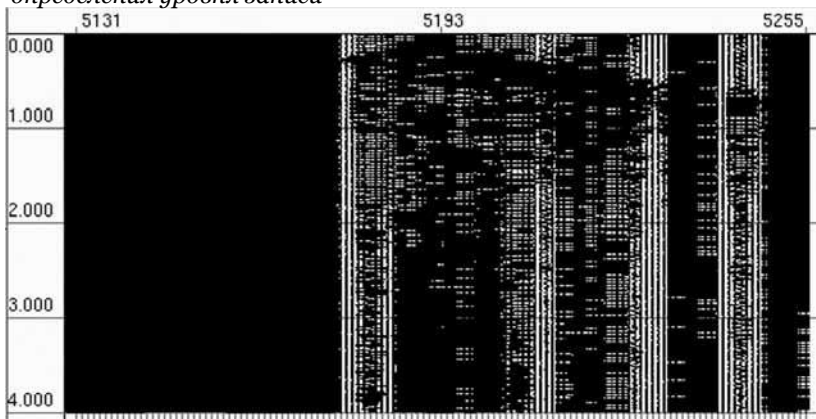


Рис. 8. Те же данные, уровень записи определен по сотой трассе.

усиления, что привело к дикому различию трасс по амплитудам. Данные, таким образом, стали совершенно непригодны для какого либо анализа истинных амплитуд, и исправить это никак нельзя, поскольку неизвестно какое усиление фактически применялось.

Проблема не возникла бы, если бы программа визуализации допускала бы масштабирование,

в зависимости от данных (как многие другие программы) или сравнение с соседними трассами (еще один стандартный способ проверки качества). Эта ошибка, возможно, весьма распространена и, по большей части, не фиксируется. Если она допущена, то она перейдет на все этапы работы, и как в рассмотренном случае, придется переформатировать большой объем данных.

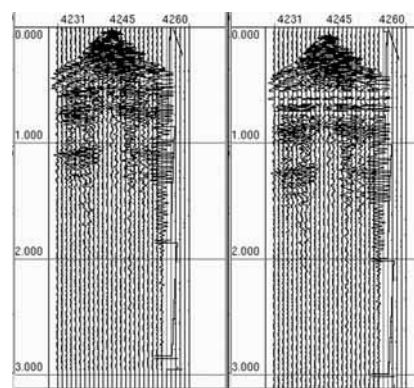


Рис. 9. Демультимплексированная полевая запись с ошибкой во времени, вызванной пропуском отсчетов (слева) и исправленная с использованием канала времени (справа).

### Проблемы при чтении

Иногда проблемы связаны с комбинацией факторов. Типичный пример приведен на рис. 9. Из-за плохого качества ленты данные были демультимплексированы не в полном объеме, а примененная программа, не обрабатывала эту ситуацию верно; в результате каждый пропущенный отсчет приводил ошибке во времени отсчета. Это хорошо видно на рисунке: один из вспомогательных каналов содержит метки времени, и видно, что за три секунды накопилась ошибка в 100 мс. Поскольку канал меток времени также был переписан, оказалось возможным исправить данные. Из-за пропущенных отсчетов результат выглядит ужасно, но отражения, по крайней мере, находятся на своих местах.

По-видимому, немало старых наземных данных были записаны с такой ошибкой, но ее можно исправить (ценой потери данных), если, конечно, был записан канал времени. Известно, однако, что по крайней мере одна крупная компания не записывает канал синхронизации, считая, что после демультимплексирования он больше не нужен!

### Выводы

Рассмотренные примеры не единичны, они все взяты изданных практических полевых работ с разными форматами записи и довольно типичны. «Человеческий фактор» особенно влияет на процесс демультимплексирования, однако ошибки в длине блока возникают

на любом этапе обработки.

Перезапись проводилась многими компаниями, и ни одна не заметила этих ошибок. Следует ожидать, что при перезаписи без проверки качества обязательно случаются ошибки, многие из которых нельзя исправить. Это, по сути, лишь верхушка айсберга.

Эти меры диктуются простым здравым смыслом, но лишь немногие компании применяют их. Поэтому владелец данных должен на этом настаивать. Тем не менее, поскольку перезапись все чаще выполняется специализированными компаниями, которые не всегда разбираются в смысле данных, с которыми работают, многие ошибки так и будут оставаться незамеченными.

Решение заключается в применении адекватных процедур проверки качества. Все трассы следует проверять на несоответствия: этот процесс можно автоматизировать; в него также нужно включать анализ данных на такой предмет, как длина записи и интервал между отсчетами, чтобы находить и объяснять отклонения от нормы. Необходимо также визуально просматривать, по крайней мере, по одному профилю из каждого перезаписываемого набора данных.

### **Благодарности**

Автор благодарит компании BP, Centrica, Rohol-Aufsuchungs, Vienna и RWE Dea, Hamburg за разрешение использовать их данные.