

Audion Media Tools — принятые решения

[English](#) · [О программе](#) · [Руководство](#) · [Замеры](#)

Содержание

- [Точность реза](#)
- [Синхрон](#)
- [Контейнеры, метаданные, пределы](#)
- [Работа и форма](#)
- [Отказы и честность](#)
- [Версии](#)
- [Решения об оболочке](#)

Что решено и почему — с датами и с числами, на которых решение стоит. Где измерение потом опровергло вывод, это тоже записано.

Точность реза

Частоту берут у файла, не угадывают — 26.08.2026

`ffprobe` сообщает точную дробь, и ничего из неё не выводится. Файл, снятый на 23,976, объявляет `24000/1001`; снятый на честные 24,000 — `24/1`. Оба читаются как написано.

Это отменяет решение от 25.08, по которому каждая целая частота NTSC читалась как её дробный близнец — на вере, будто контейнеры округляют. **Они не округляют.** ARRI Alexa Mini, снимающая настоящие 24,000 (260 кадров на 10,8333 с, измерено), объявляет `24/1`, и множитель превращал верную частоту в неверную: пять кадров расхождения за четыре минуты — та самая ошибка $\times 1,001$, ради предотвращения которой опция и существовала, только направленная в другую сторону.

Галочка убрана из панели и из обоих входов. Частота, которую не удалось прочитать, отвергается, а не заменяется догадкой.

Начало прыгает на ключевой кадр, конец считается кадрами — 25.08.2026

Начало копируемого куска может быть только ключевым кадром, поэтому оно прыгает — и настоящая точка печатается вместе с расстоянием до запрошенной. У конца такого ограничения нет: он идёт как `-frames:v N`, где N берётся из точной дроби.

`-t` измерен и отвергнут: на потоке с В-кадрами он перебирает на два-три кадра. `-avoid_negative_ts make_zero` отвергнут тоже — вместе с перемоткой до `-i` он запускает видео на 0,1 с позже звука.

Точки реза — это реальное время, таймкод остаётся меткой — 25.08.2026

Точка вводится и показывается реальным временем (`00:01:28,4`), а не таймкодом. На частотах NTSC они расходятся по определению: `00:04:00:00` без пропуска кадров на 29,97 — это кадр 7200, то есть 240,24 секунды настоящего времени. Инструмент, принимающий одно за другое, уезжает на `1001/1000` — измеренный дефект Shutter Encoder, с которого началась эта работа: он держится с версии 19.1, на 20.2 никуда не делся, четыре секунды на час.

Таймкод в операции остаётся, но только как метка, которую надо сдвинуть. Каждый перевод между секундами и кадрами идёт через точную дробь.

Рез идёт только по ключевым кадрам — 24.08.2026

Видео копируется, рез ложится на ближайший ключевой кадр. Покадровая точность — она требует перекодирования всего файла — намеренно вне области.

Обещание инструмента: быстро и без потерь. Кнопка, которая иногда работает две секунды, а иногда двадцать минут — в зависимости от галочки, которую никто не читает, — это обещание ломает. Покадровый рез умеет дюжина других программ; быть быстрым и честным об этом — более полезная ниша.

Следствие: интерфейс показывает **настоящую** точку реза, а не запрошенную — ближайший ключевой кадр со смещением словами («00:01:28,4, -1,6 с»). Тихая поправка хуже медленного реза.

Движок отвечает матрице источников, а не одному ролику — 25.08.2026

Рез проверен на россыпи камерных источников, а не на том единственном файле, на котором писался: ProRes с PCM в MOV, HEVC с длинной группой кадров, H.264 без межкадрового сжатия, 29,97 с пропуском кадров, переменная частота, четыре звуковых канала, MPEG-TS и имя с пробелами и кириллицей. Пять дефектов, которые могла показать только такая россыпь:

Кадр, стоящий чуть внутри конечной точки, принадлежит куску. Конечный индекс округляется вверх, а не к ближайшему: на 29,97 тридцатая секунда — это кадр 899,1, а кадр 899 стоит на 29,9966 — внутри. Округление к ближайшему молча его теряло.

-frames:v точен только тогда, когда перемотка попала в настоящий ключевой кадр. В любом другом положении FFmpeg считает кадры от предыдущего ключевого и отбрасывает лишние — кусок выходит короче ровно на этот зазор. Служба всегда перематывает на вычисленный ключевой кадр.

HEVC с пирамидой B-кадров пишет на два-четыре кадра меньше запрошенного. Недостача устойчива для потока, но не следует из **has_b_frames**, поэтому она измеряется, а не угадывается: кадры считаются, и если каких-то нет — рез повторяется один раз с запросом, поднятым на величину недостачи. Запрос **N + d** даёт ровно **N**.

Переменная частота запрещает арифметику кадров. **avg_frame_rate** — среднее, и счёт по нему покрывает не тот отрезок: двенадцать запрошенных секунд вышли шестнадцатью. Такой источник узнаётся по расхождению **r_frame_rate** и **avg_frame_rate**, режется по времени и помечается в журнале.

Отсчёт времени не всегда начинается с нуля. MPEG-TS обычно стартует с 1,44 с. Оператор вводит позиции от первого кадра, FFmpeg считает по собственной шкале потока. Начало вычитается из найденных ключевых кадров и прибавляется обратно при перемотке — в службе, в командном запуске и в предпросмотре.

Синхрон

Синхрон измеряется, а не предполагается — 25.08.2026

Рез, сдвинувший один поток и не сдвинувший другой, — отказ, который обнаружится последним: на монтаже, а не в журнале. Поэтому он проверяется маркером, существующим в обоих потоках в один и тот же миг: белый кадр в начале каждой секунды и щелчок 20 мс там же.

Прочитать эти маркеры правильно вышло со второй попытки. `blackdetect` и `silencedetect` работают целыми кадрами и окнами, что кладёт под измерение порог в 40 мс, а сравнение «первой замеченной вспышки» с «первым замеченным щелчком» сравнивает разные события, если один поток открывается маркером, а другой нет. Честный способ читает видео через `signalstats` с `fps_mode passthrough` — чтобы ни один кадр не дублировался под часы фильтра — и связывает каждую вспышку с её собственным щелчком по настоящим отметкам времени файла.

Измеренное так, относительно собственного смещения источника:

контейнер	расхождение с источником
MP4	0...2 мс
MOV	0...5 мс
MKV	1...3 мс
MXF	9...18 мс

Покрыты все режимы — убрать начало, убрать конец, оставить середину, разделить надвое — на AAC, на 24-битном PCM и на ProRes, включая путь с выбором канала, который пересобирает звук. Сдвигает только MXF: его муксер равняет звук по собственным единицам монтажа, это меньше половины кадра и объявлено предупреждением.

Прежнее предупреждение утверждало, что MKV гонит звук вперёд на длину звукового пакета.

Точное измерение это опровергло — MKV сдвигает начало обоих потоков вместе, а это не рассинхрон, — и предупреждение снято, а не оставлено правдоподобной страшилкой.

Контейнеры, метаданные, пределы

Настоящий камерный файл нашёл то, чего не показали синтетические — 26.08.2026

Canon `MVI_*.MP4` — HEVC Rext 4:2:2 10 бит UHD на 23,976, звук LPCM `pcm_s16be`, таймкод 15:58:57:17 — прошёл через все режимы. Кадры везде точны (242 / 240 / 166 и разделение 144 + 170, покрывающее все 314 кадров источника), таймкод сдвинулся правильно, а звук нашёлся внутри источника дословно: кусок начинается на 48 сэмплов — одну миллисекунду, сороковую долю кадра — после точки реза, скопирован, а не пережат.

Два дефекта, которые мог показать только настоящий файл:

Терялись камерные метаданные. Источник несёт производителя, модель и бренд в отдельном блоке; `-map_metadata 0` их не переносит, и муксер писал обобщённое, теряя остальное. По плану раздела камерные метаданные положено сохранять — добавлен `-movflags use_metadata_tags`, и после этого всё сверено тег за тегом.

Выбор канала менял формат сэмплов. `pcm_s16be` выходил как `pcm_s16le` — те же сэмплы с переставленными байтами и другой меткой в файле. Оба контейнера принимают PCM с обратным порядком байтов, поэтому порядок источника сохраняется.

Ни то, ни другое не всплыло бы на сгенерированных роликах: `lavfi` пишет прямой порядок байтов и никаких камерных тегов вовсе.

Чего матрица ещё не покрывает — 26.08.2026

Проверено с тех пор: две звуковые дорожки (камера плюс рекордер) обе переживают рез со сдвинутым таймкодом; метки HDR и HLG проходят без изменений; источник с открытой группой кадров режется в точное число кадров; таймкод 59,94 с пропуском кадров сдвигается верно; HEVC с LPCM в MP4 — то, что на самом деле пишут Sony и Panasonic — сохраняет `hvc1`, сохраняет `ipcm` и попадает в кадр.

Предупреждение про PCM в MP4 теперь срабатывает только при смене контейнера. Камера, снимающая HEVC с LPCM, ровно такой MP4 и пишет, и предупреждать о нём на каждом файле — шум, а не сведения.

Не проверено — и лучше сказать прямо, чем делать вид, что матрица полна: файлы длиннее часа и ряд редких сочетаний.

MXF переписывает PCM вместо копирования — 26.08.2026

Измеренное посэмплово: причина расхождения не в единицах монтажа муксера, а в том, что копирование PCM в MXF несёт лишние 384 сэмпла в начале — восемь миллисекунд.

Контейнер, который не удержит поток, отвергается до запуска — 25.08.2026

MP4 не держит ProRes, и FFmpeg говорит об этом фразой «Could not find tag for codec» и файлом нулевой длины. Известные отказы проверяются по кодеку источника заранее и сообщаются как пропуск — с причиной и с перечнем контейнеров, которые подойдут.

MPEG-TS не несёт указателя, поэтому перемотка в нём интерполируется по позициям байтов: точна в середине файла и способна промахнуться за конец рядом с ним. Это объявлено предупреждением, а результат заметно короче запланированного — или такой, что не читается вовсе — считается отказом, а не успехом со странной длительностью.

Канал, которого в файле нет, отвергается — 26.08.2026

Запрос несуществующего канала прежде давал тишину и код возврата 0.

Работа и форма

Форма показывает, что останется, а не что убирается — 25.08.2026

Форма реза отвечает на один вопрос — что останется — и отвечает наглядно. Сохраняемый отрезок подсвечен зелёным на обеих дорожках формы волны и обведён; отбрасываемое закрыто тёмной вуалью. Режимы названы по исходу («Убрать начало», «Оставить середину»), а не по действию.

Точки называются IN и OUT — словами, которыми монтажёр уже пользуется. Прежние «начало» и «конец» сталкивались со словами в названиях режимов, и на экране ничто их не различало.

Под движками — значки с подсказками, сгруппированные и выровненные как панель воспроизведения, со счётчиком кадра и времени справа. Единственный текст в строке — само поле времени, и оно объявляет свой формат: `чч:мм:сс,мс`, ни таймкод, ни сэмплы.

Две операции попали в раздел потому, что отвечают на вопросы, возникающие при выборе точки: снимок текущего кадра и разделение файла надвое в точке реза, при котором ничего не выбрасывается — две части, у второй сдвинут таймкод.

Рез идёт по одному файлу, а метки помнятся — 25.08.2026

Папка дублей не делит точки реза: у них разные ключевые кадры, разная хлопушка, разный уход из кадра. Поэтому режется один файл за раз, и файл выбирается шагами — значок с текущим именем и стрелка по бокам — а не из списка.

Точки помнятся по файлу, пока открыто окно, по дешёвому отпечатку содержимого (размер плюс первый и последний мегабайт) — переименование дубля не теряет его меток. Запуск режет каждый помеченный файл по его собственным точкам за один проход, и значок говорит, сколько их в очереди.

Прослушивание играет кадр, а не секунду — 26.08.2026

Форма волны показывает, где звук, а не что сказано, поэтому движок звучит. Кусок длиной в кадр и выровнен по границе кадра — так монтажёр и слушает: полкадра смещения это другой слог в начале слова. Есть и три кадра, когда одного мало.

Звуковая дорожка декодируется в память целиком (моно, 24 кГц), а не режется с диска на каждое движение: 0,84 мс на кусок против 0,35 с на запуск плеера. Выше трёх часов она держится блоками по две минуты — четырёхчасовая бобина заняла бы 660 МБ, а движок живёт в одной части файла за раз.

Постоянно работающий FFmpeg рассмотрен и отвергнут: запуск двоичного файла стоит 0,02 с. Стоит открытие и декодирование — именно это теперь и держит память.

Звук отвечает раньше картинки — 26.08.2026

Перерисовка прежде декодировала кадр (0,69 с) и заново рисовала форму волны (0,13 с), и только потом играла кусок ценой 0,8 мс — поиск фразы на слух означал секунду ожидания. Теперь кусок идёт первым, по более короткой задержке, а картинка догоняет своим темпом.

Тот же проход убрал две повторяющиеся траты: `ffprobe` спрашивали о неизменном файле на каждом движении, а окно формы волны перерисовывалось даже тогда, когда точка из него не выходила.

Отказы и честность

Прерванный прогон не оставляет того, что нельзя открыть — 26.08.2026

Отмена реза через полчаса работы над часовым файлом убила FFmpeg правильно и доложила об отказе правильно — и оставила 6 ГБ в папке результата под тем именем, которое носил бы готовый кусок. Муксер не успел записать указатель, поэтому файл не открывается ничем, но взгляд на папку говорит, что рез там есть.

Прежнее правило удаляло только пустые файлы — на том основании, что частичный результат может пригодиться. Это верно, пока частичный файл открывается. Проверка теперь на читаемость: что открывается — остаётся, и журнал говорит, куда дошло; что не открывается — удаляется, с причиной и размером.

Отказ перезаписать — это пропуск, а не успех — 25.08.2026

FFmpeg отвечает на  при существующей цели фразой «already exists. Exiting.» и кодом возврата 0. Прогон, не записавший ничего, читается как успех, а следующая проверка меряет предыдущий файл. Путь реза проверяет цель сам и докладывает пропуском.

Та же ловушка есть в старых путях кодирования и перепакровки — записано здесь, чтобы следующий не открывал её заново как загадку.

Версии

FFmpeg 9.0.1 проверен, а не принят на веру — 26.08.2026

Вложенная сборка перешла на 9.0.1, и каждая таблица переснята на ней: матрица контейнеров и кодеков совпала с матрицей 8.0.1 ячейка в ячейку, резы на дробных частотах легли на те же кадры, все 75 обёрток отработали без правок. Из восьми параметров, исчезнувших между версиями, здесь не используется ни один.

Какая сборка на самом деле поставляется — решает установщик по версии драйвера NVIDIA; на большинстве машин это ветка 8.x. Поэтому рабочими держатся обе, а документация называет сборку, на которой сняты числа, а не версию поставки.

Рез нацелен на камерный материал, а не на сборку фильмов — 24.08.2026

Рез сделан для камерных оригиналов: MOV и MP4, H.264, H.265 и ProRes, звук PCM или AAC, одна-две дорожки. Не для фильмов с пятью языковыми дорожками, субтитрами и AC-3 — это сборка контейнера, и другие инструменты делают её хорошо.

Следствие для набора параметров: каналы вместо дорожек (камера пишет пушку и петличку двумя моно-каналами), камерные метаданные сохраняются, а не срезаются, и никакой нормализации громкости — она уничтожает запас по уровню и связь между дублями. Нормализация принадлежит отдельному профилю для записи совещаний.

И то, что терять нельзя: **таймкод**. По нему монтажёр синхронизирует камеры между собой и с полевым рекордером. Рез обязан сдвинуть его правильно, а , как известно, молча исчезает при перепакровке MOV в MP4 — поэтому, если перенести его не удаётся, сказать об этом надо до запуска, а не после.

Решения об оболочке

Общие для всех программ Audion на этом шаблоне, приняты в мае 2026 года.

Окно — оболочка над командной строкой, а не переписывание. У инструментов уже были работающие сценарии; окно делает частые действия заметнее и безопаснее, сохраняя правду терминала.

Сервер и окно — отдельные процессы. Отдельная модель лучше пережила проверку на живой машине, избавляет от сюрпризов с браузером и упрощает разбор конфликтов за порт.

Тёмная раскладка, терминал в главной роли. Слева подготовка, папки, действия и параметры; справа состояние и журнал, занимающий около двух третей высоты. Вывод этих программ не второстепенен — он часть работы.

Компактные прозрачные кнопки вместо крупных залитых: у будущих инструментов бывает по два десятка команд, и большие кнопки съедают высоту и ломаются на длинных русских подписях. Подписи команд выравниваются влево — центрирование обрезает длинную подпись с обеих сторон, пряча начало.

Две колонки держатся примерно до 900 точек ширины и остаются пригодными на ноутбуке 1920×1200 при масштабе 150 %. Окно стартует в 1600×900 при минимуме около 1180×720, с перетаскиваемым разделителем.

Порядок полей следует решениям пользователя, а не порядку доводов командной строки. Первый видимый блок отвечает на вопрос «чем и на чём это будет выполнено», редкое прячется в раскрывающийся блок, состояние которого запоминается.

Одно видимое действие на один результат. Никаких «аудит» и «весь процесс» рядом, никаких двух механизмов избранного на одном экране. Действие обязано называть свой объект: «Ключ в избранное», а не просто «В избранное».

Внешний ввод переносится к себе. Кнопки добавления копируют выбранное во входную папку: безопаснее, чем долгие операции по сетевым путям и съёмным дискам.

Итоговое состояние постоянно. Под журналом держится индикатор: серый, синий, зелёный, красный. Всплывающее сообщение можно не заметить, если окно было неактивно.

Помощники командной строки скрываются при создании процесса — флагами Python, а не только ключом PowerShell: окно успевает мелькнуть раньше, чем ключ подействует.

Кодировка `.cmd` — **условие сборки**. Все файлы проекта в UTF-8 без метки порядка байтов и со строгими CRLF; проверка встроена в сборку, установку, диагностику и упаковку выпуска, а не живёт устными напоминаниями.