

Обрезка: подробно

Раздел режет камерные файлы **без перекодирования**. Потоки копируются как есть, поколение не теряется, метаданные и таймкод переезжают в результат. Это не конкурент монтажной программе, а точный отсекатель лишнего с гарантией, что материал с камеры доедет до монтажа таким, каким снят.

Всё, что здесь написано, проверено прогоном на живых файлах; числа в тексте — измеренные, а не ожидаемые. Свод измерений — в `MEASUREMENTS_RU.md`, таблицы совместимости — в `TRIM_MATRIX_RU.md`, клавиши плеера — в `MPV_HOTKEYS_RU.md`.

Рабочий цикл

1. Положить дубли в `Source`.
2. Выбрать файл бейджем — стрелки слева и справа шагают по папке, кнопка с таблицей показывает, что `ffprobe` знает о файле, ещё до запуска.
3. `Открыть в плеере` — дубль открывается в mpv отдельным окном.
4. Найти точки: `,` и `.` шагают ровно в кадр, `Shift+←` и `Shift+→` — точно на секунду, `l` ставит A, второе нажатие — B. Кусок между точками играет по кругу, и рез слышно до того, как он сделан.
5. `Взять A и B` — обе точки переезжают в поля панели. Или `Отметить IN` и `Отметить OUT`, если удобнее держать руки на панели: они берут позицию, где стоит плеер, и переносят туда же точку петли.
6. Выбрать, что сделать с куском между точками, и нажать `Запустить`.

Точки можно и просто набрать с клавиатуры: `00:01:28,400` или просто секунды. Способ, которому не нужно ничего, кроме цифры, — и он работает, даже если плеер не установлен.

Что сделать с куском между точками

Три кнопки. Больше не нужно: какой край уходит, уже сказано тем, какие поля заполнены.

Оставить

Остаётся то, что между точками, остальное уходит.

Заполнено	Что происходит
только IN	уходит начало: хлопушка, подход, фальстарт
только OUT	уходит конец: отход и остановка камеры
обе точки	уходят оба края, остаётся середина
ничего	раздел отказывается: «нечего резать, точка не задана»

Вырезать

Наоборот: кусок между точками исчезает, а начало и конец склеиваются в один файл. Оговорка, телефонный звонок, прошедший через кадр ассистент.

Работает в три прохода FFmpeg, ни один из них не перекодирует: голова до IN, хвост от ключевого кадра после OUT, склейка копированием пакетов. Промежуточные куски живут в `workspace` и удаляются, чем бы дело ни кончилось.

Разделить надвое

Точка одна, ничего не выбрасывается: получаются `имя_part1` и `имя_part2`.

Почему начало ложится на ключевой кадр

Копируемый видеопоток нельзя начать с произвольного кадра: кадры между ключевыми описаны через соседей, и без опорного кадра их нечем декодировать. Поэтому начало куска ложится на ближайший ключевой кадр, а журнал говорит, куда оно попало на самом деле:

```
cut in 00:00:09.000 -> keyframe 00:00:08.400 (-0.6 s)
```

Конец считается иначе — кадрами по точной дроби частоты, поэтому попадает ровно в запрошенный кадр:

```
cut out 00:00:21.000 -> frame 300 of the piece (00:00:12.000)
result 300 frames, 00:00:12.000 (+0.000 s from the plan), timecode 01:00:09:00
```

У выреза середины это правило работает наоборот. Хвост начинается с **первого ключевого кадра после OUT**, а не с ближайшего: начнись он раньше — и часть того, что просили убрать, осталась бы в результате. Насколько больше запрошенного ушло, сказано словами:

```
tail at 00:00:09.000 (first keyframe past the piece, +0.7 s)
[WARNING] 0.7 s more than asked for was removed. Starting earlier would leave
part of the removed piece in the result.
```

Если ключевой кадр стоит ровно в точке — не уходит ничего лишнего.

Частота берётся у файла, а не додумывается

`ffprobe` отдаёт точную дробь: `24000/1001` для 23,976 и `24/1` для честных 24,000 — камера снимает и так, и так. Дробь идёт в расчёт как дробь, поэтому на часовом файле нет накопленной ошибки.

Если файл снят с **переменной частотой** (телефон пишет 30 в одном поле и что-то среднее в другом), кадровый счёт соврал бы: раздел переходит на счёт по времени и говорит об этом. Конец тогда попадает внутрь пакета, а не в кадр.

Звук

Звук копируется вместе с картинкой. Два случая, где копия недостаточно точна, и раздел переписывает PCM тем же форматом и той же разрядностью — побитово тот же звук, но нарезанный по сэмплу:

- **MXF**: чанки большие, копия притащила бы около 8 мс звука с той стороны реза;
- **склейка** (**Вырезать**): излишек не просто повисает в конце, он приходится ровно на стык. Измерено на дубле 25 к/с: без этого стык уезжал на 72 мс, потом на 16 мс; сейчас — ровно 1 200 000 сэмплов на 25,000 с.

Аудиоканал умеет взять первый, второй или свести оба в моно — камера часто пишет пушку и петличку двумя моно-каналами. Выбор канала **пересобирает** звук: для PCM это бесплатно, для сжатого стоит поколения, и раздел об этом предупреждает.

Что ещё едет вместе с материалом

Камера пишет не только картинку и звук, и раньше всё остальное отбрасывалось молча — человек узнавал об этом уже в монтаже. Теперь это два чекбокса, оба включены по умолчанию.

Телеметрия — поток данных. GoPro пишет туда свою телеметрию. Такой поток держат **MOV и MP4; MKV не держит вовсе** («Only audio, video, and subtitles are supported for Matroska»), MXF тоже нет — и раздел говорит об этом до запуска, а не падает на середине.

Субтитры — у камерных файлов редкость, но DJI пишет субтитрами данные полёта. Их принимают MOV, MP4 и MKV.

Дорожка таймкода — тоже поток данных, и она намеренно остаётся: FFmpeg не может назвать её кодек, MP4 на такой останавливается («Could not find tag for codec none»), а сам таймкод и так переезжает значением. В журнале это сказано словами, с именем дорожки:

```
[WARNING] The data stream (tmcd) has no codec FFmpeg can name, so it cannot be
        copied. A timecode track reads exactly like this and is written as a
        value instead.
```

Что переехало — пишется тоже:

```
carried over: telemetry, subtitles
```

Правило общее для панели и командной строки (**AUDION_TRIM_DATA**, **AUDION_TRIM_SUBTITLES**), и файлы у них по-прежнему совпадают побайтово.

Таймкод

Монтажёры синхронизируют камеры между собой и с рекордером по таймкоду, поэтому он переезжает в результат и сдвигается на длину отрезанного начала:

```
timecode 01:00:00:00 -> 01:00:09:00
```

Ноль вместо **Сдвиг** обнуляет таймкод результата. Если контейнер не умеет держать **tmcd**, раздел скажет об этом до запуска, а не после.

Когда раздел откажет — до запуска

Случай	Что говорит
Камерный RAW в MXF (Sony X-OCN, ARRIRAW)	FFmpeg не читает такой поток вовсе — резать нечего; названа программа производителя
MP4 + ProRes, DNxHR, FFV1, HuffYUV, UTVideo, v210, CineForm	контейнер не примет такое видео; предложены MOV или MKV
MOV + AV1 или VP9	то же самое
MXF + HEVC, AV1, VP9	MXF таких не держит
MXF + сжатый звук	MXF принимает только PCM
MOV + FLAC, Opus, Vorbis	контейнер не примет такой звук
Запрошен канал, которого нет	сказано, сколько каналов в файле на самом деле
Обе точки пусты	«нечего резать: точка не задана»

Читаемые расширения: **avi**, **m2ts**, **m4v**, **mkv**, **mov**, **mp4**, **mts**, **mxv**, **ts**.

Плеер

mpv запускается **отдельной программой** и слушает команды по каналу — так же, как здесь используется **ffmpeg.exe**. Ставится один раз: **install\Install-Portable-mpv.cmd**.

Что это даёт:

- точная перемотка: попадание в запрошенный кадр, а не в ближайший ключевой;
- шаг ровно в кадр в обе стороны;
- воспроизведение **назад** (**b**), чего не умеет ни один браузерный проигрыватель;

- петля A/B — она и есть рез, слышимый до реза;
- те же камерные форматы, что читает сама программа: внутри mpv тот же FFmpeg.

Окно можно держать на втором мониторе; оно переживает смену файла — переоткрытие это команда, а не перезапуск. Если закрыть его руками, панель это увидит и кнопка **Открыть в плеере** поднимет его заново. Вместе с программой плеер закрывается сам.

Установщик кладёт рядом с плеером текст лицензии (**LICENSE.GPL**, **COPYRIGHT** — в архиве сборки их нет) и исходники, из которых он собран: сам mpv по коммиту сборки, встроенный в него FFmpeg — это **другой** FFmpeg, не тот, которым работает программа, — и сборочные скрипты, где перечислены версии всех остальных библиотек. Около 24 МБ; что откуда взято, записано в **Tools\mpv\SOURCES.txt**.

Настройки лежат в **config\mpv**. Там конфигурация владельца из Audion TC слово в слово, а ниже — короткий блок поправок для реза: интерполяция выключена (она рисует кадры, которых в файле нет, и точку по ним ставить нельзя), перемотка всегда точная, файл открывается туда, куда его открыли.

Клавиши — в **MPV_HOTKEYS_RU.md**.

Из командной строки

Scripts\ff-trim-start.cmd	убрать начало
Scripts\ff-trim-end.cmd	убрать конец
Scripts\ff-trim-both.cmd	оставить середину
Scripts\ff-trim-middle.cmd	вырезать середину и склеить

Точки задаются переменными окружения: **AUDION_TRIM_FILE**, **AUDION_TRIM_START**, **AUDION_TRIM_END**; контейнер, звук, канал и таймкод — **AUDION_TRIM_CONTAINER**, **AUDION_TRIM_AUDIO**, **AUDION_TRIM_AUDIO_CHANNEL**, **AUDION_TRIM_TIMECODE**.

Арифметика общая с панелью, и это проверено, а не заявлено: один и тот же вырез, сделанный обёрткой и панелью, дал **побайтово одинаковые файлы**.

Соседний раздел: разворот без потерь

Разворот картинки живёт в [Упаковка / Remux](#), потому что это не рез, а смена того, как файл показывать. Камеру повернули набок ради вертикального кадра, а она записала обычный горизонтальный файл и ничего об этом не сказала — здесь разворот записывается в контейнер, и монтажка открывает клип правильно.

Пиксели не трогаются: файл 6 085 307 байт стал 6 085 304, хэши кадров совпали, декодер отдаёт 540×960 вместо 960×540. Хранят разворот MP4, MOV и MKV; MXF нет, и раздел отказывает вместо того, чтобы выдать неповёрнутый файл. Предлагается для камерных H.264 и HEVC.