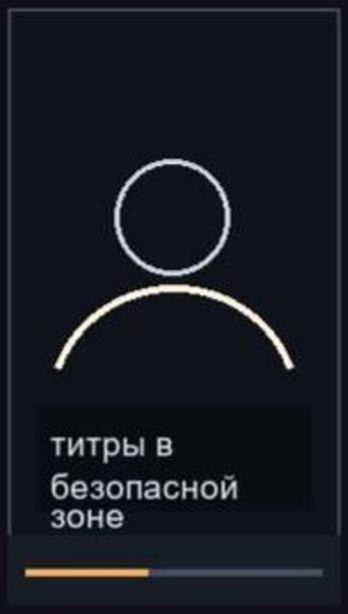
FFMPEG · PILLOW · ЗВУК

Вертикальный ролик скриптом

покадровая анимация, 9:16 и -14 LUFS

Рендер в pipe без временных файлов, rim-свет дугой, субтитры отдельным SRT и контроль по таймкодам



720 x 1280

24 fps · h264

AAC 48 кГц

-14 LUFS

TP -1.2 · LRA 11

1,4 МБ — вложением

```
-af "loudnorm=I=-14:TP=-1.2" -ar 48000
-f rawvideo -i - -crf 18 -preset medium
```

ЛИС практическая серия

Шадрин Данил

Вертикальный ролик скриптом: 2D-анимация, вырезка 9:16 и звук на слабом ПК

«Автор»

2026

Данил Ш. Б.

Вертикальный ролик скриптом: 2D-анимация, вырезка 9:16 и звук на слабом ПК / Ш. Б. Данил — «Автор», 2026

Как я собираю вертикальные ролики не в монтажной программе, а скриптами: покадровая 2D-анимация, вырезка 9:16, контровой свет по контуру головы, субтитры в безопасной зоне и звук, приведённый к стандарту минус 14 LUFS. Всё это на ноутбуке с видеокартой на 4 ГБ и 23 ГБ оперативной памяти, где тяжёлый редактор живёт плохо, а поштучный расчёт кадров работает отлично. Внутри живые команды ffmpeg с разбором каждого флага, таблица спецификации трёх готовых роликов, которую можно перепроверить утилитой ffprobe, код отрисовки силуэта и дуги контрового света, генерация звуковой волны чистым питоном без единого стороннего файла и объяснение, почему рендер разбит на части и как потом склеивается. Отдельно честная оговорка: код, сцены и музыка мои, текст книги написан при помощи ИИ-агента, клиентских кейсов я не выдумываю.

© Данил Ш. Б., 2026
© Автор, 2026

Содержание

Вертикальный ролик скриптом 5
Покадровая 2D-анимация, вырезка 9:16 и звук до −14 LUFS на видеокарте с 4 ГБ 5
От автора 5
Глава 1. Конечная точка: что должно получиться 5
Глава 2. Рендер: кадры летят в ffmpeg по pipe 6
Конец ознакомительного фрагмента. 7

Шадрин Данил
Вертикальный ролик скриптом: 2D-анимация, вырезка 9:16 и звук на слабом ПК

Вертикальный ролик скриптом

Покадровая 2D-анимация, вырезка 9:16 и звук до −14 LUFS на видеокарте с 4 ГБ

От автора

Мне девятнадцать, я учусь на последнем курсе и монтирую вертикальные ролики не в монтажной программе, а скриптами. Это не причуда и не попытка «заехать с другой стороны»: у меня ноутбук с видеокартой на 4 ГБ и 23 ГБ оперативной памяти, и тяжёлый редактор на нём живёт плохо, а расчёт на кадры — отлично.

Код, сцены, позы, музыка и вся сборка — мои. Текст книги как таковой написан при помощи ИИ-агента, с которым я работаю каждый день: он черновик, я читаю и правлю. Клиентских кейсов я тут выдумывать не буду, их нет; будут реальные параметры с моих файлов, и их можно перепроверить_ffprobe_ом.

Глава 1. Конечная точка: что должно получиться

Вот спецификация трёх готовых роликов, снятая с файлов, а не из головы:

Файл
Длительность
Кадр
Частота
Видео
Аудио
Размер

vertical_1_rain.mp4
13,0 с
720×1280
24
h264, ~1 196 кбит/с
AAC 48 кГц, стерео
1,94 МБ

vertical_2_comet.mp4
14,0 с

720×1280
24
h264, ~818 кбит/с
AAC 48 кГц, стерео
1,43 МБ

vertical_3_final.mp4
14,0 с
720×1280
24
h264, ~845 кбит/с
AAC 48 кГц, стерео
1,48 МБ

Размер меньше двух мегабайт выбран не «для экономии»: такой файл можно приложить прямо в переписку, и он не упрётся в лимиты мессенджера. Это часть монтажа, а не упаковка.

Глава 2. Рендер: кадры летят в ffmpeg по pipe

Никаких тысяч png-шек на диске. draw_frame(t) рисует кадр в памяти, и он уходит в стандартный вход ffmpeg как raw video:

```
ffmpeg = subprocess.Popen([
"ffmpeg", "-y", "-f", "rawvideo", "-pix_fmt", "rgb24", "-s", f"{W}x{H}",
"-r", str(FPS), "-i", "-", "-an", "-c:v", "libx264", "-preset", "medium",
"-crf", "18", "-pix_fmt", "yuv420p", "-movflags", "+faststart", str(out),
], stdin=subprocess.PIPE, stdout=subprocess.DEVNULL, stderr=subprocess.DEVNULL)

for i in range(f0, f1):
t = i / FPS
frame = draw_frame(t).convert("RGB")
ffmpeg.stdin.write(frame.tobytes())
```

Что тут важнее, чем кажется:

-i - со rawvideo вместо последовательности файлов. Диск не участвует, а на машине, где системный диск был забит под ноль, это вопрос выживания.

-an на первом проходе. Видеоряд без звука: звук прилетает позже отдельным слоем, иначе любая правка реплики означает перерендер картинки.

-crf 18 и -preset medium. Ниже 18 на 4 ГБ смысл теряет скорость, выше — заметно сыпятся градиенты ночного неба, а они у меня как раз ночные.

+faststart. Без него плеер не начнёт показывать ролик, пока не скачает конец файла, а на телефоне это выглядит как «видео не открывается».

Конец ознакомительного фрагмента.

Текст предоставлен ООО «Литрес».

Прочитайте эту книгу целиком, купив полную легальную версию на Литрес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.