

Аппаратные кодеки VP9

АО "МЦСТ"

16 ноября 2023 г.

1 Введение

На момент написания кодеки Google G2 (декодер) и Google BigE (энкодер) используются в процессорах E2C3 и R2000+. Для поддержки аппаратных кодеков должны быть установлены библиотеки кодеков (пакет `omx-il-vp9dec`). При корректной работе аппаратные кодеки будут использоваться по-умолчанию при попытке кодирования/декодирования VP9-видео.

1.1 Декодирование VP9

Поддержка аппаратного декодера реализована в VLC, GStreamer (через `gst-omx`), FFmpeg, Firefox (через FFmpeg). В VLC аппаратный декодер используется как при декодировании VP9 напрямую, так и при использовании библиотеки `libavcodec` (FFmpeg). Первый вариант является предпочтительным.

VLC

```
vlc -V xcb_xv input.webm
```

VLC (libavcodec)

```
vlc -V xcb_xv input.webm --codec avcodec
```

GStreamer (xvimagesink)

```
gst-launch-1.0 filesrc location=input.webm ! matroskademux ! omxvp9dec ! xvimagesink
```

GStreamer (filesink)

```
gst-launch-1.0 filesrc location=input.webm ! matroskademux ! omxvp9dec ! \  
filesink location=output.yuv
```

FFmpeg

```
ffmpeg -i input.webm -c:v rawvideo -pix_fmt nv12 output.yuv
```

- Диапазон поддерживаемых входных разрешений: 180x180 — 4096x2304.
- Шаг входного разрешения — 16 пикселей.
- Аппаратный выходной формат декодера — NV12, поддерживается программная конвертация в I420.

Для работы аппаратного декодирования в Firefox должна быть отключена опция `media.rdd-vpx.enabled` в `about:config`.

1.2 Кодирование VP9

Поддержка аппаратного энкодера реализована в GStreamer (через `gst-omx`) и FFmpeg.

GStreamer

```
gst-launch-1.0 filesrc location=input.yuv ! rawvideoparse width=640 height=480 \
format=i420 framerate=30/1 ! omxvp9enc target-bitrate=1152000 mode=offline effort=5 ! \
webmmux ! filesink location=output.webm
```

FFmpeg

```
ffmpeg -f rawvideo -pix_fmt yuv420p -s:v 640x480 -r 30 -i input.yuv -c:v vp9enc_omx \
-b:v 1152000 -mode offline -effort 5 output.webm
```

2 Доступные опции кодирования

Для сжатия кадров энкодер применяет квантование. Величина квантования контролируется параметром квантования (QP). Увеличение параметра квантования QP ведёт к ухудшению качества видео и к лучшему сжатию видеопотока (уменьшение битрейта). Значение QP варьируется от 1 до 255. При QP < 10 обычно очень трудно увидеть разницу между оригинальным и сжатым видео.

2.1 Управление битрейтом

- **control-rate** — Параметр задаёт режим управления битрейтом и пропуск кадров:
 - **Покадровое управление битрейтом** — Покадровый алгоритм управления битрейтом. Если включен, то значение QP вычисляется на каждом кадре. В выключенном состоянии при кодировании используется одно фиксированное значение QP для всех кадров (без учёта буста).
 - **Пропуск кадров** — Если итоговый битрейт превышает заданный и не может быть уменьшен лишь средствами алгоритма покадрового управления битрейтом, энкодер может начать пропускать кадры. Пропуск кадров работает только в однопроводных режимах без задержки:
`mode=none ref-frame-scheme=0 (или 1,2,3,5) lookahead-count=0 arf-temporal-filter-enabled=0 passes=1 pass-number=0`

Режим пропуска кадров выбирается опцией `skip-mode`. При использовании пропуска кадров энкодер кодирует оставшиеся кадры с меньшим значением QP.

Значение	Управление битрейтом	пропуск кадров
disable	–	–
variable (default)	+	–
constant	+	–
variable-skip-frames	+	+
constant-skip-frames	+	+

GStreamer

```
gst-launch-1.0 filesrc location=input.yuv ! rawvideoparse width=640 height=480 \
format=i420 framerate=30/1 ! omxvp9enc target-bitrate=1152000 \
control-rate=constant mode=offline effort=5 ! webmmux ! filesink \
location=output.webm
```

FFmpeg

```
ffmpeg -f rawvideo -pix_fmt yuv420p -s:v 640x480 -r 30 -i input.yuv \
-c:v vp9enc_omx -b:v 1152000 -control-rate constant -mode offline -effort 5 \
output.webm
```

- **skip-mode = 0..4 [0]** — Режим пропуска кадров. Работает только при включенном пропуске кадров, т.е. при `control-rate=variable-skip-frame/constant-skip-frames`, и только при `ref-frame-scheme ≠ 4` и `lookahead-count = 0`.

Значение	Описание
0 (default)	Автоматический режим
1	Пропуск 1 из 5 кадров
2	Пропуск 1 из 3 кадров
3	Пропуск 1 из 2 кадров
4	Пропуск 2 из 3 кадров

- **qp-hdr = -1..255 [-1]** — Значение QP по-умолчанию или исходное значение QP для энкодера. Оно используется при кодировании первого кадра, если включено покадровое управление битрейтом, либо при кодировании всех кадров, если покадровое управление битрейтом выключено. Не рекомендуется использовать значения меньше 10. Значение -1 означает выставление данного параметра энкодером на основании разрешения, фреймрейта и целевого битрейта.

Покадровое управление битрейтом выключено:

GStreamer

```
gst-launch-1.0 filesrc location=input.yuv ! rawvideoparse width=640 height=480 \
format=i420 framerate=30/1 ! omxvp9enc target-bitrate=1152000 \
control-rate=disable mode=lowlatency effort=1 qp-hdr=40 ! webmmux ! filesink \
location=output.webm
```

FFmpeg

```
ffmpeg -f rawvideo -pix_fmt yuv420p -s:v 640x480 -r 30 -i input.yuv \
-c:v vp9enc_omx -b:v 1152000 -control-rate disable -mode lowlatency -effort 1 \
-qp-hdr 40 output.webm
```

Покадровое управление битрейтом включено:

GStreamer

```
gst-launch-1.0 filesrc location=input.yuv ! rawvideoparse width=640 height=480 \
format=i420 framerate=30/1 ! omxvp9enc target-bitrate=1152000 \
control-rate=variable mode=lowlatency effort=1 qp-hdr=40 ! webmmux ! filesink \
location=output.webm
```

FFmpeg

```
ffmpeg -f rawvideo -pix_fmt yuv420p -s:v 640x480 -r 30 -i input.yuv \
-c:v vp9enc_omx -b:v 1152000 -control-rate variable -mode lowlatency -effort 1 \
-qp-hdr 40 output.webm
```

- **qp-min = 0..255 [0]** — Минимальное значение QP, которое может быть использовано энкодером. Не рекомендуется использовать значения меньше 10.
- **qp-max = 0..255 [0]** — Максимальное значение QP, которое может быть использовано энкодером.
- **target-bitrate = 10000..60000000 [1000000]** — Целевой битрейт в битах в секунду, используется когда включено управление битрейтом.
- **gop-length = 1..300 [150]** — Окно битрейта. Алгоритм управления битрейтом управляет битрейтом внутри группы кадров, называемой окном битрейта. Алгоритм использует окно битрейта для подстройки среднего битрейта каждого окна к целевому битрейту. Таким образом, данный параметр влияет на то, как быстро алгоритм управления битрейтом реагирует на изменения в видеопотоке (например, скачки битрейта закодированных кадров). Окно битрейта в 150 кадров при кодировании видеопотока 30fps будет подстраивать итоговый битрейт в течение 5 секунд. Следует учитывать, что если `golden-picture-rate > 0`, то `gop-length = min(3*golden-picture-rate, gop-length)`.

- **intra-qp-delta = -127..127 [-20]** — Коррекция QP для Intra-кадров. Наличие Intra-кадров в видеопотоке VP9 может вызывать заметное мерцание из-за разных методов предсказания для Intra- и Inter-кадров. Эту проблему можно решить, скорректировав параметр квантования Intra-кадров относительно окружающих их Inter-кадров. По-умолчанию энкодер использует значение -20 .
- **fixed-intra-qp = 0..255 [0]** — Фиксированный QP для Intra-кадров. Параметр заставляет энкодер использовать заданное фиксированное значение QP для каждого Intra-кадра.
- **intra-picture-rate = 0..+∞ [0]** — Частота Intra-кадров. Параметр определяет частоту, с которой энкодер обновляет Intra-кадры.
- **golden-picture-rate = 0..+∞ [10]** — Частота Golden Frames. Параметр определяет частоту, с которой энкодер обновляет Golden Frames. Параметр применим только при кодировании в реальном времени, т.е. при `ref-frame-scheme` $\neq 4$ и `lookahead-count` = 0.
- **adaptive-golden-update = 0..1 [1]** — Адаптивное обновление Golden Frames. Энкодер проверяет использование Golden Frame в тот момент, когда его нужно обновить. Если Golden Frame используется часто и включено адаптивное обновление Golden Frames, энкодер может не обновлять Golden Frame.
- **adaptive-golden-boost = 0..100 [40]** — Адаптивный буст для Golden Frames. Golden Frames используются как опорные кадры для многих последующих кадров. Кодирование Golden Frames в более высоком качестве позволяет последующим кадрам также получить заметное улучшение в качестве за небольшой штраф в виде увеличившегося размера Golden Frame. Параметр определяет величину буста в % для каждого Golden Frame относительно Intra Frame. Параметр применим только при `ref-frame-scheme` $\neq 4$.

В некоторых случаях, при использовании экстремально низкого битрейта, энкодер может не достичь его независимо от того, насколько высокие значения QP использовать. В этой ситуации можно использовать пропуск кадров.

Для ограничения алгоритма управления битрейтом можно устанавливать минимальное и максимальное значения QP, тогда алгоритм будет использовать QP только из заданного отрезка. Это полезно в режиме переменного битрейта, когда постоянный уровень качества предпочтительнее постоянного битрейта. Однако эти ограничения ухудшают возможности алгоритма управления битрейтом по достижению целевого битрейта.

2.2 Параметры кодирования

- **mode** — Режим кодирования. Доступно 4 предустановленных режима кодирования: `lowlatency` (default), `realtime`, `lagged`, `offline`.
 - **Двухпроходный режим (offline, lagged)** — Двухпроходный режим генерирует видео наилучшего качества. Можно запустить два отдельных прохода кодирования, либо кодирование с задержкой.

При кодировании в два отдельных прохода энкодер сначала проходит через всё видео и, основываясь на собранной статистике, принимает решения, как видео будет закодировано во время второго прохода. Двухпроходный режим обычно конфигурируется для работы в режиме переменного битрейта, чтобы поддерживалось постоянное качество картинки и больше бит расходовалось на значимую информацию.

При кодировании с задержкой энкодер перед вторым проходом сначала выполняет первый проход для N кадров. Статистика собирается аналогично полноценному двухпроходному режиму, но, так как первый проход не прошёл предварительно через всю последовательность, кодирование может быть не таким эффективным.
 - **Режим реального времени с задержкой (realtime)** — Режим реального времени с задержкой используется в задачах, которые могут выиграть от наличия временных зависимостей в видео, и для которых допустима некоторая задержка кодирования. Такими задачами являются, например, локальная запись видео, либо его трансляция. В зависимости от задачи, алгоритм управления битрейтом можно сконфигурировать для кодирования с переменным или постоянным битрейтом. В режиме кодирования с постоянным битрейтом качество изображения зависит от сложности видеопоследовательности.
 - **Быстрый режим реального времени (lowlatency)** — Данный режим используется для задач, требующих минимальную задержку, таких как видео-конференции и трансляция экрана. Управление битрейтом обычно настраивается на режим постоянного битрейта. Так как нет

предварительного прохода и алгоритм управления битрейтом не знает ничего о будущем, выходной битрейт может отклоняться от целевого.

Режим кодирования жёстко определяет параметры `lookahead-count`, `arf-temporal-filter-enabled`, `passes`, `pass-number`, `ref-frame-scheme`. При необходимости все эти параметры можно переопределить, используя режим `none`.

mode	lookahead-count	arf-temporal-filter-enabled	passes	pass-number	ref-frame-scheme
lowlatency	0	false	1	0	0
realtime	12	false	1	0	4
lagged	12	false	2	0..2 [0]	4
offline	20	true	2	0..2 [0]	4
none	0..25 [0]	false, true [false]	0..2 [1]	0..2 [0]	0..5 [0]

Выбрать номер прохода `pass-number` можно только в режимах `lagged`, `offline` и `none`.

Для ручного режима `none` действуют ограничения: при `lookahead-count = 0` `temporal-filter` принудительно устанавливается в 0. При `passes = 2` минимальное значение `lookahead-count = 6`. При `ref-frame-scheme = 4` минимальное значение `lookahead-count = 2`.

- **effort = 0..5 [0]** — Сила сжатия. Аппаратура энкодера способна оценивать несколько векторов движения, размеров преобразований и фильтровать кандидатов через цикл RDO. При силе сжатия 0-3 каждый дополнительный уровень сжатия удваивает время, затрачиваемое на кодирование, но улучшает эффективность сжатия. Уровень 4 дополнительно добавляет к уровню 3 перекодирование кадра с оптимальными значениями вероятностей. Уровень 5 добавляет перекодирование с использованием оптимальной силы фильтра.
- **passes = 1, 2 [2]** — Общее число проходов при кодировании. Один проход — 1, два прохода — 2.
- **pass-number = 0, 1, 2 [0]** — Номер текущего прохода. Оба прохода — 0, один проход — 1, два прохода — 2.
- **interpolation-filter = 0..4 [4]** — Определяет тип интерполяционного фильтра, используемого для компенсации движения: 0 (smooth), 1 (normal), 2 (sharp), 3 (bilinear) или 4 (switchable). Если выбран `switchable`, то энкодер сам выбирает оптимальный тип фильтра.
- **filter-level = 0..64 [64]** — Определяет силу деблокирующего фильтра. Большая сила означает больше фильтрации, т.е. более размытое изображение. Автоматическая настройка позволяет энкодеру самому выбрать силу на основе значения QP. Значение 64 — высчитывать автоматически во время кодирования.
- **filter-sharpness = 0..8 [8]** — Определяет резкость фильтра. Автоматическая настройка позволяет энкодеру самому выбрать резкость на основе значения QP. Значение 8 — высчитывать автоматически во время кодирования.
- **error-resilient = false, true [false]** — Включает помехоустойчивое кодирование. Уменьшает влияние ошибок в битах и потерянных пакетов, но генерирует поток большего размера.
- **frame-parallel = false, true [false]** — Параллельное декодирование кадров. Включает поддержку параллельного декодирования кадров. Например, отключает обратную адаптацию вероятностей.
- **quality-metric = psnr, ssim [psnr]** — Метрика качества. Параметры выбора режима кодирования оптимизированы для двух разных метрик: PSNR или SSIM.
- **qp-delta-ydc, qp-delta-ch-dc, qp-delta-ch-ac = -15..16 [0]** — Коррекция QP. Базовое значение QP используется для коэффициентов luminance AC и по-умолчанию это же значение QP используются для всех преобразованных коэффициентов. Коррекция QP позволяет тонко настраивать значения QP отдельно для блоков luminance DC, luminance 2-го порядка и chrominance.
- **high-prec-mv-enable = 0..1 [1]** — Высокоточные векторы движения. Использование точности в 1/8 пикселя для векторов движения позволяет достичь лучшей эффективности сжатия по-сравнению с точностью в 1/4 пикселя. Если включено, энкодер сам решает, использовать высокоточные векторы движения, или нет.

- **me-chroma-weight = 0..3 [2]** — Вес chrominance в алгоритме оценки движения. $0 = 1x$, $1 = 2x$, $2 = 4x$, $3 = 8x$.
- **lossless = false, true [false]** — Сжатие без потерь. Выставляет все параметры фильтров и параметр квантования в ноль.
- **lookahead-count = 0..25 [20]** — Число кадров, на которое второй проход отстаёт от первого при кодировании с задержкой. Для входных форматов RGBA/BGRA с разрешением более 2560x1440 максимальное значение lookahead-count ограничено 15 кадрами.
- **ref-frame-scheme = 0..5 [4]** — Схема обновления опорных кадров. Стандарт VP9 предусматривает наличие трёх опорных кадров: Last, Golden и AltRef. Last всегда является предыдущим закодированным кадром.
 - **0 (Auto)** — В этом режиме схема кодирования выбирается автоматически из приведённых ниже.
 - **1 (Last three)** — В этом режиме AltRef — это предыдущий Golden, а Golden — это предыдущий Last.
 - **2 (Static Golden AltRef 2nd Last)** — В этом режиме AltRef — это предыдущий Last, а Golden следует с частотой golden-picture-rate.
 - **3 (Static Golden static AltRef)** — В этом режиме AltRef — это предыдущий Golden, до первого Golden поведение не определено.
 - **4 (Towoway)** — Схема описана в таблице.
 - **5 (Static Golden smart AltRef)** — В этом режиме Golden совпадает с AltRef, обновляется когда разница между Last и Golden/AltRef превышает 2, тогда Golden = AltRef = Last

При ref-frame-scheme = 0 схема кодирования может меняться динамически во время кодирования.

- **Временная фильтрация Alt-Ref Frames:**

- **arf-temporal-filter-enabled = true, false [false]** – Включает временную фильтрацию. Фильтр также включается автоматически при `lookahead-count ≠ 0`.
- **arf-temporal-filter-strength = 0..15 [6]** – Определяет силу временного фильтра. Значение 0 отключает фильтрацию.
- **arf-temporal-filter-length = 0..1 [1]** – Определяет число последовательных кадров, используемых при временной фильтрации (3 или 6 кадров). Значение 1 приводит к более агрессивной временной фильтрации, 0 - к менее агрессивной.
- **arf-temporal-filter-threshold = 0..64000 [1000]** – Определяет то, насколько фильтр чувствителен к изменениям (например, к движению объектов) в последовательных кадрах. Высокие значения параметра приводят к более агрессивной фильтрации с меньшим вниманием к временным изменениям (низкая чувствительность к движению). Низкие значения приводят к менее агрессивной фильтрации с большим вниманием к временным изменениям и лучшему сохранению движущихся деталей (высокая чувствительность к движению).

- **cq-level = 0..255 [0]** – Определяет минимальное значение QP для обычных кадров. Параметр применим только в двухпроходных режимах.
- **min-arf-period** – Минимальный период AltRef Frames. Определяет минимальную дистанцию между AltRef Frames. Должен быть меньше `lookahead-count - 2`. Применим только в режиме `ref-frame-scheme = 4`
- **max-arf-period** – Максимальный период AltRef Frames. Определяет максимальную дистанцию между AltRef Frames. Должен быть \geq `min-arf-period`. Применим только в режиме `ref-frame-scheme = 4`

2.3 Параметры препроцессора

Препроцессор может выполнять одну операцию: либо поворот и отражение, либо обрезку и даунскейлинг видео.

- **rotation = 0, 90, 180, 270 [0]** – Поворот изображения. Блок поворота поворачивает изображение на ± 90 градусов, либо на 180 градусов. При использовании поворота высота изображения должна быть кратна 8.

GStreamer

```
gst-launch-1.0 filesrc location=input.yuv ! rawvideoparse width=640 height=480 \
format=i420 framerate=30/1 ! omxvp9enc target-bitrate=1152000 mode=offline \
effort=5 rotation=180 ! webmmux ! filesink location=output.webm
```

FFmpeg

```
ffmpeg -f rawvideo -pix_fmt yuv420p -s:v 640x480 -r 30 -i input.yuv \
-c:v vp9enc_omx -b:v 1152000 -mode offline -effort 5 -rotation 180 output.webm
```

- **flip = none, vert [none]** – Отражение изображения по вертикали. При использовании отражения высота изображения должна быть кратна 8.

GStreamer

```
gst-launch-1.0 filesrc location=input.yuv ! rawvideoparse width=640 height=480 \
format=i420 framerate=30/1 ! omxvp9enc target-bitrate=1152000 mode=offline \
effort=5 flip=vert ! webmmux ! filesink location=output.webm
```

FFmpeg

```
ffmpeg -f rawvideo -pix_fmt yuv420p -s:v 640x480 -r 30 -i input.yuv \
-c:v vp9enc_omx -b:v 1152000 -mode offline -effort 5 -flip vert output.webm
```

- **crop = crop_left,crop_width:crop_top,crop_height** — Обрезка видео. Чтобы обрезать левые 100px видео 640x480: crop=100,540:0,480. Обрезка по 80px с каждой стороны: crop=80,480:80,320.

GStreamer

```
gst-launch-1.0 filesrc location=input.yuv ! rawvideoparse width=640 height=480 \
format=i420 framerate=30/1 ! omxvp9enc target-bitrate=1152000 mode=offline \
effort=5 crop=80,480:80,320 ! webmmux ! filesink location=output.webm
```

FFmpeg

```
ffmpeg -f rawvideo -pix_fmt yuv420p -s:v 640x480 -r 30 -i input.yuv \
-c:v vp9enc_omx -b:v 1152000 -mode offline -effort 5 -crop 80,480:80,320 \
output.webm
```

- **scale = scaled_width x scaled_height** — Масштабирование видео. Поддерживается только даунскейлинг. Максимальный коэффициент даунскейлинга 1/16 для каждого измерения. Доступен только для входного формата I420.

GStreamer

```
gst-launch-1.0 filesrc location=input.yuv ! rawvideoparse width=640 height=480 \
format=i420 framerate=30/1 ! omxvp9enc target-bitrate=1152000 mode=offline \
effort=5 crop=80,480:80,320 scale=240x160 ! webmmux ! filesink location=output.webm
```

FFmpeg

```
ffmpeg -f rawvideo -pix_fmt yuv420p -s:v 640x480 -r 30 -i input.yuv \
-c:v vp9enc_omx -b:v 1152000 -mode offline -effort 5 -crop 80,480:80,320 \
-scale 240x160 output.webm
```

- **color-conversion = 0, 1, 2, 3 [0]** — Тип преобразования из RGB в YUV. Используется только для входных форматов RGBA, BGRA.

Значение	Тип преобразования
0 (default)	BT.601, studio swing [16..235]
1	BT.709, studio swing [16..235]
2	BT.601, full swing [0..255]
3	BT.709, full swing [0..255]

GStreamer

```
gst-launch-1.0 filesrc location=input.rgb4 ! rawvideoparse width=640 height=480 \
format=rgba framerate=30/1 ! omxvp9enc target-bitrate=1152000 mode=offline \
effort=5 color-conversion=3 ! webmmux ! filesink location=output.webm
```

FFmpeg

```
ffmpeg -f rawvideo -pix_fmt rgba -s:v 640x480 -r 30 -i input.rgb4 -c:v vp9enc_omx \
-b:v 1152000 -mode offline -effort 5 -color-conversion 3 output.webm
```

- **firstpass-scale = 0..100 [100]** — Часть кадра (в %), обрабатываемая в первом проходе. Возможно ускорить выполнение первого прохода, подавая для него уменьшенный кадр или некоторую часть кадра.
- **scale-method = scale, crop [scale]** — Метод уменьшения кадра для обработки в первом проходе. Метод scale доступен только для входного формата I420.

2.4 Поддерживаемые входные форматы

Энкодером поддерживаются разрешения входного видео от 136x96 до 4096x2304. В качестве входных для энкодера поддерживаются следующие форматы:

FourCC	GStreamer	FFmpeg	width (min)	width (max)	Δ width (max)
I420	i420	yuv420p	192	4096	64
NV12	nv12	nv12	160	4096	32
NV21	nv21	nv21	160	4096	32
YUY2	yuy2	yuyv422	144	4096	16
UYVY	uyvy	uyvy422	144	4096	16
RGBA	rgba	rgba	136	4096	8
BGRA	bgra	bgra	136	4096	8

Высота (height) входного изображения должна быть чётной. Для входных видео в форматах RGBA/BGRA с разрешением более 2560x1440 максимальное значение lookahead-count ограничено 15 кадрами.

Пример кодирования видео входного формата BGRA:

GStreamer

```
gst-launch-1.0 filesrc location=input.bgra ! rawvideoparse width=640 height=480 \
format=bgra framerate=30/1 ! omxvp9enc target-bitrate=1152000 mode=offline effort=5 ! \
webmmux ! filesink location=output.webm
```

FFmpeg

```
ffmpeg -f rawvideo -pix_fmt bgra -s:v 640x480 -r 30 -i input.bgra -c:v vp9enc_omx \
-b:v 1152000 -mode offline -effort 5 output.webm
```

2.5 Примеры

Двухпроходное кодирование с постоянным QP:

GStreamer

```
gst-launch-1.0 filesrc location=input.yuv ! rawvideoparse width=640 height=480 \
format=i420 framerate=30/1 ! omxvp9enc mode=offline effort=5 passes=2 pass-number=1 \
control-rate=disable qp-hdr=20 ! webmmux ! filesink location=output.webm
```

```
gst-launch-1.0 filesrc location=input.yuv ! rawvideoparse width=640 height=480 \
format=i420 framerate=30/1 ! omxvp9enc mode=offline effort=5 passes=2 pass-number=2 \
control-rate=disable qp-hdr=20 ! webmmux ! filesink location=output.webm
```

FFmpeg

```
ffmpeg -f rawvideo -pix_fmt yuv420p -s:v 640x480 -r 30 -i input.yuv -c:v vp9enc_omx \
-mode offline -effort 5 -passes 2 -pass-number 1 -control-rate disable -qp-hdr 20 \
output.webm
```

```
ffmpeg -f rawvideo -pix_fmt yuv420p -s:v 640x480 -r 30 -i input.yuv -c:v vp9enc_omx \
-mode offline -effort 5 -passes 2 -pass-number 2 -control-rate disable -qp-hdr 20 \
output.webm
```

Двухпроходное кодирование с постоянным битрейтом:

GStreamer

```
gst-launch-1.0 filesrc location=input.yuv ! rawvideoparse width=640 height=480 \  
format=i420 framerate=30/1 ! omxvp9enc target-bitrate=1152000 mode=offline effort=5 \  
passes=2 pass-number=1 qp-min=10 qp-max=160 ! webmmux ! filesink location=output.webm
```

```
gst-launch-1.0 filesrc location=input.yuv ! rawvideoparse width=640 height=480 \  
format=i420 framerate=30/1 ! omxvp9enc target-bitrate=1152000 mode=offline effort=5 \  
passes=2 pass-number=2 qp-min=10 qp-max=160 ! webmmux ! filesink location=output.webm
```

FFmpeg

```
ffmpeg -f rawvideo -pix_fmt yuv420p -s:v 640x480 -r 30 -i input.yuv -c:v vp9enc_omx \  
-b:v 1152000 -mode offline -effort 5 -passes 2 -pass-number 1 -qp-min 10 -qp-max 160 \  
output.webm
```

```
ffmpeg -f rawvideo -pix_fmt yuv420p -s:v 640x480 -r 30 -i input.yuv -c:v vp9enc_omx \  
-b:v 1152000 -mode offline -effort 5 -passes 2 -pass-number 2 -qp-min 10 -qp-max 160 \  
output.webm
```

Двухпроходное кодирование с задержкой и с постоянным битрейтом:

GStreamer

```
gst-launch-1.0 filesrc location=input.yuv ! rawvideoparse width=640 height=480 \  
format=i420 framerate=30/1 ! omxvp9enc target-bitrate=1152000 mode=offline effort=5 \  
qp-min=10 qp-max=160 ! webmmux ! filesink location=output.webm
```

FFmpeg

```
ffmpeg -f rawvideo -pix_fmt yuv420p -s:v 640x480 -r 30 -i input.yuv -c:v vp9enc_omx \  
-b:v 1152000 -mode offline -effort 5 -qp-min 10 -qp-max 160 output.webm
```

Быстрое двухпроходное кодирование с задержкой и с постоянным битрейтом:

GStreamer

```
gst-launch-1.0 filesrc location=input.yuv ! rawvideoparse width=640 height=480 \  
format=i420 framerate=30/1 ! omxvp9enc target-bitrate=1152000 mode=lagged effort=3 \  
qp-min=40 qp-max=200 ! webmmux ! filesink location=output.webm
```

FFmpeg

```
ffmpeg -f rawvideo -pix_fmt yuv420p -s:v 640x480 -r 30 -i input.yuv -c:v vp9enc_omx \  
-b:v 1152000 -mode lagged -effort 3 -qp-min 40 -qp-max 200 output.webm
```

Кодирование в реальном времени с задержкой и с постоянным битрейтом:

GStreamer

```
gst-launch-1.0 filesrc location=input.yuv ! rawvideoparse width=640 height=480 \  
format=i420 framerate=30/1 ! omxvp9enc target-bitrate=1152000 mode=realtime effort=1 \  
qp-min=60 qp-max=240 ! webmmux ! filesink location=output.webm
```

FFmpeg

```
ffmpeg -f rawvideo -pix_fmt yuv420p -s:v 640x480 -r 30 -i input.yuv -c:v vp9enc_omx \  
-b:v 1152000 -mode realtime -effort 1 -qp-min 60 -qp-max 240 output.webm
```

Быстрое кодирование с постоянным битрейтом:

GStreamer

```
gst-launch-1.0 filesrc location=input.yuv ! rawvideoparse width=640 height=480 \  
format=i420 framerate=30/1 ! omxvp9enc target-bitrate=1152000 mode=lowlatency effort=0 \  
intra-picture-rate=20 gop-length=30 qp-min=100 qp-max=255 ! webmmux ! filesink \  
location=output.webm
```

FFmpeg

```
ffmpeg -f rawvideo -pix_fmt yuv420p -s:v 640x480 -r 30 -i input.yuv -c:v vp9enc_omx \  
-b:v 1152000 -mode lowlatency -effort 0 -intra-picture-rate 20 -gop-length 30 \  
-qp-min 100 -qp-max 255 output.webm
```