

차세대 동영상 코덱 압축 효율 비교: HEVC vs VP9

김일구
삼성전자
ilkoo.kim@samsung.com

Coding Efficiency Comparison between Next Generation Video Codecs:
HEVC vs VP9

Il-Koo Kim
Samsung Electronics Co., Ltd.

요 약

본 논문에서는 JCT-VC 에서 2013 년 1 월에 표준화가 완료된 High Efficiency Video Coding (HEVC)과 구글에서 2013 년 6 월에 개발 완료 예정인 VP9 의 압축 효율 비교를 수행한다. HEVC 는 UHD 등 고화질 방송 등에 대응하도록 디자인 되었으며, VP9 은 유튜브 (YouTube) 등과 같은 인터넷 비디오 스트리밍에 적합하도록 디자인되었다. VP9 의 경우 HEVC 와는 달리 로열티 프리 (royalty-free)를 지향하며 오픈소스 (open source) 방식으로 개발이 진행되고 있다. 본 논문에서는 HEVC 와 VP9 의 디자인 차별점을 소개하고, 랜덤 액세스 환경 (Random Access, RA)과 저지연 환경 (Low Delay, LD)에서 HEVC 와 VP9 의 압축 효율을 비교한다. 실험 결과에 따르면, 방송 및 패키지 미디어 등에서 많이 사용될 랜덤 액세스 환경에서는 VP9 이 HEVC 대비 32.7% 열세를 보인다. 비디오 컨퍼런스등과 같은 저지연 환경에서는 VP9 이 HEVC 대비 26.7% 열세를 보인다. VP9 의 경우 개발이 완료된 것이 아니므로, 향후 압축 효율의 향상이 있을 것으로 기대된다.

1. 서론

MPEG-2 대비 2 배의 압축 효율을 갖는 H.264/AVC [1] [2]가 표준화가 완료된 이래, 멀티미디어 데이터에 대한 수요는 멈추지 않고 지속적으로 증가하고 있다. 이에 대응하기 위해 ITU-T Video Experts Group (VCEG)과 ISO/IEC MPEG 에서는 Joint Collaboration Team on Video Coding (JCT-VC)을 결성하여, 2010 년 초부터 H.264/AVC 대비 2 배의 압축 효율을 목표로 High Efficiency Video Coding (HEVC)에 대한 표준화를 시작하였다 [3] [4]. 그 결과로, 2013 년 1 월에 첫번째 (version 1) 표준규격을 완료하였다. 현재는 첫번째 표준의 범위 확장 (Range extension), 스케일러블 (scalable) 비디오 및 3D 비디오 등과 같은 확장 표준에 대한 표준화가 진행되고 있다.

이와 별도로, 구글에서는 인터넷 비디오 스트리밍에 특화된 동영상 코덱을 지속적으로 개발 및 공개하고 있다. 2010 년에는 로열티 프리 (royalty-free)를 지향하며 오픈 소스 (open source) 방식으로 개발한, VP8 을 개발 및 공개하였고, 그 후속 버전인 VP9 의 개발을 2013 년 6 월에 완료할 예정이다 [5] [6] [7]. VP8 의 경우 H.264/AVC 와 동등한 압축 효율을 가진다고 알려져 있으며, 현재 개발이 진행되고 있는 VP9 의 목표는 인터넷 스트리밍 환경에서 HEVC 와 동등 수준의 압축 효율을 갖는 것이다. 본 논문에서는 차세대 동영상 코덱으로 널리 사용될 두 코덱에 대해 차별점을 소개하고,

다양한 환경에서 압축 효율을 비교 분석한다.

본 논문의 구성은 다음과 같다. 2 절에서는 두 코덱의 디자인 차별점들을 살펴본 후, 3 절에서는 HEVC 와 VP9 의 압축 효율을 다양한 환경에서 실험을 통해서 비교 분석한다. 마지막으로 4 절에서는 본 논문에 대한 결론을 맺는다.

2. 기술 개요

2.1 블록 구조 (block structure)

HEVC 와 VP9 모두 기존의 H.264/AVC 및 VP8 보다 큰 예측 블록 (prediction block)을 사용한다. 압축의 기본단위가 되는 블록을 HEVC 에서는 코딩 유닛 (Coding Unit)이라고 하고 VP9 에서는 Super Block (SB)이라고 한다. 두 코덱은 64x64 크기를 갖는 코딩 유닛 (Coding Tree Unit, CTU) 혹은 SB64 에서 부터 시작하여 쿼드트리 (quad-tree) 분할을 사용하여 4x4 크기를 갖는 예측 블록까지 지원할 수 있다. HEVC 에서는 쿼드트리의 모든 노트에서 동일한 분할 방법을 사용하여 4x4 블록까지 분할하지만 VP9 의 경우 SB64, SB32 (32x32 예측 블록)를 제외한 MB (16x16 예측 블록) 이하는 기존의 VP8 의 분할 방법을 재사용한다는 점이 크게 다른 점이다. SB64 와 SB32 의 경우 각 SB 가 인트라 예측 모드 혹은 인터 예측 모드를 가진다.

2.2 인트라 예측 (intra prediction)

HEVC의 경우 DC, Planar 와 함께 33 개의 기술기 (angular) 모드를 포함하여 총 35 개의 인트라 예측 모드를 갖는데 반해 VP9의 경우 그 보다 적은 10 개의 인트라 예측 모드를 갖는다. VP9의 경우에는 4x4 인트라 예측의 경우 VP8 과 동일한 모드를 이용해서 예측하며, 8x8 보다 큰 예측 블록의 경우는 새롭게 정의된 10 개의 인트라 예측 모드를 사용하여 예측한다. 8x8 이상 블록에서 사용되는 인트라 예측 모드는 다음과 같다: DC, Vertical, Horizontal, TM, and 6 기술기 예측 모드 (기술기: 27, 45, 63, 117, 135, 153).

2.3 인터 예측 (inter prediction)

VP9의 경우 최대 3 개의 프레임 버퍼들이 존재한다. LAST_FRAME 은 현재 프레임의 바로 이전 프레임을 나타낸다. GOLDEN_FRAME 은 현재 프레임보다 과거의 임의의 프레임을 나타낸다. ALTREF_FRAME 은 GOLDEN_FRAME 을 중심으로 과거 혹은 미래 프레임들을 이용하여 생성한 프레임을 나타낸다. ALTREF_FRAME 의 경우 참조 프레임으로 사용되지만, 디스플레이되지 않는다. 현재 프레임에 대해서 8 개의 프레임들을 갖는 집합에서 3 개의 가용한 프레임을 선택하여 사용한다. 프레임 헤더를 파싱하면 어떤 프레임들이 참조되었는지 알 수 있다.

VP9에서는 lag-in-frames 과 GOLDEN_FRAME, ALTREF_FRAME 의 개념을 활용하여 양방향 예측의 효과를 낼 수 있다. 아래 그림은 양방향 예측 효과를 설명하는 그림이다.

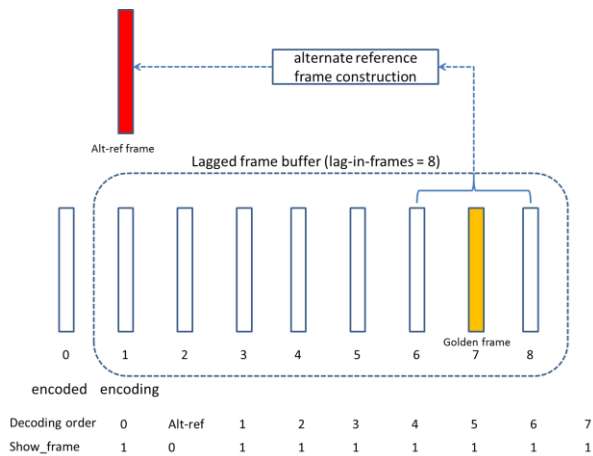


Figure 1. lag-in-frames 과 GOLDEN_FRAME, ALTREF_FRAME 의 개념을 활용하여 양방향 예측 효과를 설명하는 그림

7 번 프레임이 GOLDEN_FRAME 으로 결정되면, 주위의 프레임을 이용해서 ALTREF_FRAME 을 생성할 수 있다. ALTREF_FRAME 은 1 번 프레임이 인코딩 되기 전에 코딩되지만, 디스플레이 되지 않는다. 이후의 프레임들은 ALTREF_FRAME 을 참조하여 코딩할 수 있다. GOLDEN_FRAME 의 경우 ALTREF_FRAME 과의 차분만을 코딩한다.

하나의 움직임 벡터를 사용할 경우 지정된 참조 프레임을 보간 (interpolation) 하여 예측한다. 두 개의 움직임 벡터를 사용할 경우, 두개의 참조 프레임은 항상 달라야 하며 각 움직임 벡터를 사용하여 구한 두개의 예측을 평균하여

예측으로 사용한다. 인터-인트라 결합 예측 모드 (Compound INTER-INTRA Mode)에서는 인터 예측과 인트라 예측을 결합하여 사용한다. 인트라 예측의 모서리 부분 (블록의 위쪽 혹은 왼쪽)에서는 인트라 예측에 더 강한 웨이트 (weight)를 할당하고, 반대의 경우에는 인터 예측 블록에 더 강한 웨이트를 할당한다. 인터-인트라 결합 예측 모드는 하나의 움직임 벡터를 사용하는 16x16 이상의 블록 (MB/SB32/SB64)에서만 사용한다.

2.4 참조 프레임 보간 (Sub-pel interpolation)

HEVC의 경우 1/4-픽셀 정확도를 갖는 참조 프레임 보간 방법을 사용한다. 휘도 (luma) 성분의 경우 1/2-픽셀 위치의 경우 8-tap 필터를 사용하여 보간하고, 1/4-픽셀 위치인 경우는 7-tap 필터를 사용하여 보간한다. 색차 (chroma) 성분의 경우 4-tap 필터를 사용하여 보간한다. 휘도 및 색차 성분의 보간에 사용하는 필터는 모두 DCT-IF (DCT-based Interpolation Filter) 디자인 방법을 사용하여 생성되었다. DCT-IF의 경우 높은 주파수를 잘 보존하는 성질이 있다.

VP9은 1/8-픽셀 정확도를 갖는 참조 프레임 보간 방법을 예측에 사용한다. 1/4-픽셀 정확도 혹은 1/8-픽셀 정확도를 사용할지는 인코더에서 프레임 레벨에서 결정하여 디코더에 전송한다. 일반적으로 1/8-픽셀 정확도는 이전 프레임과 움직임 벡터가 작은 경우에 사용한다. 보간 필터의 경우 프레임 레벨 혹은 MB 레벨에서 결정가능한 3 개의 8-tap 보간 필터를 사용한다: 8-tap 일반 필터, 8-tap DCT-IF, 8-tap smooth 필터. 8-tap smooth 필터의 경우 서브-픽셀 위치의 보간에 이용되는 것이 아니라 정수-픽셀 위치의 픽셀을 smoothing 하는데 사용한다.

2.5 움직임 벡터 예측 및 코딩 (motion vector prediction and coding)

현재 블록의 주위에 있는 움직임 벡터를 우선순위 결정 방법에 따라 정렬된 리스트를 생성하고, best, nearest, near 의 세가지 후보 벡터들을 결정한다. 우선 순위의 결정에는 현재 블록의 주위 픽셀 값들을 이용한 템플릿 정합 (template matching)을 사용한다. best, nearest, near 움직임 벡터 중에서 템플릿 사이의 분산 (variance)을 비교하여 최종적인 리스트가 생성된다.

인코더에서는 nearest 혹은 near 움직임 벡터가 사용되면 관련 정보들을 전송하고, new 움직임 벡터가 사용되면 best 후보와의 차이가 offset 정보로 디코더에 전송된다. 또한, (0, 0) 움직임 벡터를 가지는 zeromv 가 사용될 수 있다. 움직임 벡터 예측을 위해 VP8에서는 현재 블록의 주위에 있는 움직임 벡터들만을 사용하였으나, VP9에서는 참조 프레임에서 현재 블록과 동일 위치에 존재하는 블록 (co-located block)의 움직임 벡터도 사용된다.

2.6 공간 변환 (transform)

HEVC와 VP9 모두 4x4, 8x8, 16x16, 32x32 크기의 DCT 기반 공간 변환을 사용하여 압축을 수행한다. 예외적으로, HEVC는 4x4 크기의 인트라 예측 모드의 잔차신호 (residual)에 대해서 DCT 대신, DST (Discrete Sine Transform)를 사용한다. VP9의 경우 특정한 인트라 모드와 결합하여 ADST (Asymmetric Discrete Sine Transform)를 사용할 수 있다. 예를 들면, 왼쪽 에지 (edge)를 이용하여 인트라 예측을 수행하는 모드의 잔차신호는 가로 방향 ADST와 세로방향 DCT를 사용하여 변환한다. TM_PRED와

대각선 방향 예측의 경우에는 가로, 세로 모두 ADST 를 사용한다.

2.7 엔트로피 코딩 (entropy coding)

HEVC 는 기존 H.264/AVC 에서 사용되는 CABAC (Context Adaptive Binary Arithmetic Coding)을 계승하였으며, 병렬화 및 저복잡도를 위해 컨텍스트 (context)의 수를 감소시키고 확률 갱신을 스킵할 수 있는 바이패스빈 (by-pass bin)의 수를 증가시켰다. VP9 은 개념적으로 HEVC 와 동일한 이진산술부호화방법 (binary arithmetic coding)를 사용한다. VP9 의 산술부호기는 8-bit 정확도를 갖는 확률 변수를 사용한다.

2.8 루프 필터 (loop filter)

HEVC 와 VP9 의 디블록 필터는 모두 다양한 크기의 예측 및 변환 블록에 대응하는 경계 결정 방법이 사용된다. 결정된 경계에 따라 평탄도 탐지 (flatness detector) 방법 및 평탄 영역이 달라지며 디블록 필터의 강도 및 길이가 결정된다.

HEVC 에는 VP9 과 달리 SAO (Sample Adaptive Offset)가 루프필터로 추가적으로 사용된다.

3. 실험 결과

압축 효율 비교를 위해서 세계의 코덱 (VP8, VP9, HM10)을 사용하였다. VP8 과 VP9 의 경우 git 저장소 (git repository)에서 2013 년 5 월 8 일 버전을 내려받아 실험에 사용하였다. HEVC 의 경우 HEVC software repository 에서 HM10 을 다운로드하여 비트스트림을 생성하였다. 실험 환경은 기본적으로 HEVC 표준화에 사용된 공통실험조건 (common test condition)을 기준으로 하였다. [8] 압축 효율의 수치와를 위해 BD-rate 을 사용하였다. [9]

3.1 VP8 과 VP9 의 압축 효율 비교

가장 먼저 VP8 과 VP9 의 압축 효율을 랜덤 액세스 환경에서 비교한다. 랜덤 액세스 환경을 구현하기 위해, kf-max-dist 입력 매개변수는 입력 영상의 프레임율 (frame rate)에 따라서 최소 1 초에 한장씩 키 프레임 (key frame, intra frame)이 삽입되도록 설정하였다. 또한 8 프레임의 버퍼 지연을 갖게 하기 위해서 lag-in-frames 입력 매개변수는 8 로 설정하였다.

Table 1. VP8 과 VP9 의 압축 효율 비교

VP8 vs VP9	랜덤 액세스 환경		
	Y	U	V
Class A (8bit)	-28.7%	-35.2%	-38.1%
Class B	-39.5%	-42.2%	-43.7%
Class C	-29.4%	-29.9%	-30.6%
Class D	-23.2%	-25.0%	-24.3%
Class E	-46.7%	-54.3%	-56.2%
Overall	-34.3%	-37.6%	-38.7%

Table 1에서 알 수 있듯이 VP9 은 VP8 대비 크게 압축 효율이 높음을 알 수 있다. 수치적으로 보면, VP8 대비 -34.3%의 BD-rate 을 갖는다.

3.2 랜덤 액세스 환경에서 HM10 과 VP9 의 압축 효율 비교

이번 절에서는 HM10 과 VP9 의 압축 효율을 랜덤 액세스 환경에서 비교한다. HM10 의 경우, 공통실험조건에 따라 인트라 삽입 주기 (intra period)를 수정하여 약 1 초 간격으로 랜덤 액세스 포인트를 삽입하도록 설정하였다. 인트라 삽입 주기는 입력 비디오의 프레임율에 따라 설정한다. 마찬가지로, VP9 의 경우 kf-max-dist 입력 매개변수를 입력 비디오의 프레임율에 따라 설정하여 키 프레임 삽입 주기를 1 초 간격으로 설정한다. 참고적으로 실험에 사용된 VP9 의 비트스트림을 생성하기 위한 명령어는 appendix 에 제시되어 있다.

HM10 의 경우 각 입력 비디오별로 공통실험조건에 정의된 4 개의 QP 들을 사용하여 비트스트림을 생성하였다. VP9 의 경우는 정확한 압축 효율 비교를 위해서 (BD-rate 의 경우 비교하는 두개의 비트율의 차이가 5% 이내일때 가장 정확한 결과를 구해준다.) 2-패스 비트율 제어 (2-pass rate control)을 사용하여 HM10 에서 생성한 비트율을 타겟 비트율로 설정하여 비트스트림을 생성하였다.

또한 HM10 의 GOP 를 8 로 세팅하고 VP9 의 경우 lag-in-frames 를 8 로 세팅하여 비트스트림을 생성하였다. 이러한 실험 설정에서는 HM10 의 경우 시간적 계층 (temporal layer) 별로 다른 QP 가 할당되는 계층적 B (hierarchical B) 구조가 사용되고, VP9 의 경우 lag-in-frames 입력 매개변수는 8 로 설정되어 ALTREF_FRAME 을 이용한 양방향 예측의 효과를 사용할 수 있다.

참고로 본 실험에서는 HEVC 공통실험조건인 Class B 의 경우 BasketballDrive 와 BQTerrace 비디오를 제외한 3 개의 시퀀스를 사용하였다.

Table 2. HM10 과 VP9 의 압축 효율 비교

HM10 vs VP9	랜덤 액세스 환경		
	Y	U	V
Class A (8bit)	26.9%	29.1%	26.4%
Class B*	41.9%	43.7%	45.9%
Class C	25.6%	21.2%	22.5%
Class D	29.8%	22.5%	23.9%
Class E	40.9%	23.1%	21.9%
Overall	32.7%	28.0%	28.8%

Table 2은 랜덤 액세스 환경에서 HM10 이 생성한 비트스트림의 크기를 기준으로 해서 VP9 이 생성한 비트스트림의 상대적 크기를 보여주고 있다. 표에서 알 수 있듯이 VP9 코덱은 HM10 대비 평균 32.7%의 BD-rate 을 갖는것을 알 수 있다.

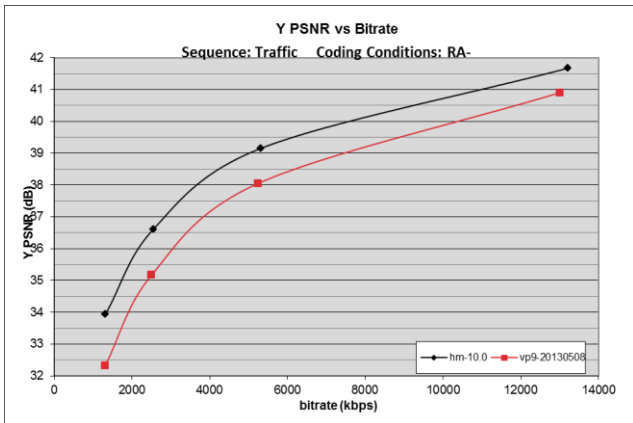


Figure 2. Rate-distortion curve of Traffic sequences

Figure 2은 테스트 비디오 중 하나인 Traffic 영상의 비트율-왜곡 (rate-distortion) 커브를 보여 주고 있다. HM10 의 경우 VP9 대비 동일 비트율에서 약 1 dB 정도 PSNR 이 높은 것을 확인 할 수 있다. 참고적으로 Traffic 의 BD-rate 은 38.8%이다.

3.3 저지연 (low delay) 환경에서 HM10 과 VP9 의 압축 효율 비교

저지연 환경에서의 성능 비교를 위해 HM 의 경우 공통실험조건인 LDP 환경을 사용하여 인트라 프레임에 입력 비디오의 맨 처음 한장만 코딩하고 나머지는 B 혹은 P 슬라이드로 코딩하였다. VP9 의 경우에는 kf_max_dist 입력 매개 변수를 조정하여 키 프레임이 중간에 삽입되지 못하게 막아주었다. 또한 저지연 환경 구성을 위해 lag-in-frames 입력 매개 변수를 0 으로 설정하였다.

Table 3. HM10 과 VP9 의 압축 효율 비교

HM10 vs VP9	저지연 환경		
	Y	U	V
Class A (8bit)	30.3%	41.9%	30.9%
Class B	28.3%	21.4%	17.4%
Class C	12.5%	-2.8%	0.5%
Class D	20.0%	-3.2%	-2.5%
Class E	51.4%	9.8%	7.6%
Overall	26.7%	5.7%	5.3%

Table 3는 저지연 환경에서는 HM10 이 생성한 비트스트림의 크기를 기준으로 해서 VP9 이 생성한 비트스트림의 상대적 크기를 보여주고 있다. 표에서 알 수 있듯이 VP9 코덱은 HM10 대비 평균 26.7%의 BD-rate 을 갖는 것을 알 수 있다.

본 실험 결과를 해석함에 있어서 주의할 점은 각 코덱의 인코딩-디코딩 복잡도가 고려되지 않았다는 점이다. 또한 HM 의 경우 전체 시퀀스에 동일한 양자화 매개변수 (quantization parameter, QP)를 사용하였으며, VP8 과 VP9 의 경우 2-패스 비트율 제어를 사용하여 비트율을 HM10 과 비슷하게 조정하였다.

4. 결론

본 논문에서는 HEVC 와 VP9 의 다양한 코딩틀을 소개하고, 다양한 환경에서 HEVC 코덱과 VP9 코덱의 압축 성능을 비교하였다. 방송등에서 많이 사용될 랜덤 액세스 환경에서는 VP9 이 HEVC 대비 32.7%의 BD-rate 을 갖는다. 비디오 컨퍼런스등과 같은 저지연 환경에서는 VP9 이 HEVC 대비 26.7%의 BD-rate 을 갖는다. VP9 의 경우 2013 년 6 월에 개발이 완료될 예정임으로, 향후 압축 효율의 향상을 기대할수 있을 듯 하다.

References

- [1] Advanced Video Coding for Generic Audio-Visual Services, ITU-T Rec. H.264 and ISO/IEC 14496-10 (AVC), ITU-T and ISO/IEC JTC 1, May 2003 (and subsequent editions).
- [2] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, A. Luthra, "Overview of the H.264/AVC Video Coding Standard," *IEEE Transactions on Circuits and Systems for Video Technology* Vol. 13 No. 7, Jan 2011.
- [3] B. Bross, W.-J. Han, G. J. Sullivan, J.-R. Ohm, and T. Wiegand, *High Efficiency Video Coding (HEVC) Text Specification Draft 9*, document JCTVC-K1003, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), Oct. 2012.
- [4] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Trans. on Circuits and Systems for Video Tech.*, Vol. 22, No. 12, Dec 2012.
- [5] J. Bankoski, J. Koleszar, L. Quillio, J. Salonen, P. Wilkins, Y. Xu, VP8 Data Format and Decoding Guide, RFC 6386
- [6] A. Grange and H. Alvestrand, "A VP9 Bitstream Overview", *Internet-Draft*, February 2013.
- [7] J. Bankoski, R. S. Bultje, A. Grange, Q. Gu, J. Han, J. Koleszar, D. Mukherjee, P. Wilkins and Y. Xu, "Towards a Next Generation Open-source Video Codec", February 2013, *IS&T / SPIE EI Conference on Visual Information Processing and Communication IV*
- [8] F. Bossen, *Common Conditions and Software Reference Configurations*, document JCTVC-L1006, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Geneva, Jan. 2013.
- [9] G. Bjøntegaard, "Calculation of average PSNR differences between RD curves," VCEG-M33, 13th VCEG meeting, Austin, Texas, March 2001.

Appendix

VP9 비트스트림을 생성하기 위한 명령어 예시

```

vpxenc -o seq_name.vp9 seq_name.yuv --codec=vp9 -w 416 -h 240 -p 2 -t 4 --best --end-usage=vbr --auto-alt-ref=1 --lag-in-frames=8 --verbose --psnr --target=29.97 --target-bitrate=30000 --minsection-pct=5 --maxsection-pct=800 --kf-min-dist=0 --kf-max-dist=32 --static-thresh=0 --drop-frame=0 --min-q=0 --max-q=60
    
```