



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

**Análise de qualidade do Low Complexity
Enhancement Video Coding (LCEVC) baseado em
AVC e VVC**

Lucas Araújo Pena

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Orientador
Prof. Ricardo L. de Queiroz

Coorientador
Prof. Eduardo Peixoto Fernandes da Silva

Brasília
2025



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Análise de qualidade do Low Complexity Enhancement Video Coding (LCEVC) baseado em AVC e VVC

Lucas Araújo Pena

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Prof. Ricardo L. de Queiroz (Orientador)
CIC/UnB

Prof. Camilo Chang Dórea Prof. Edson Mintsu Hung
CIC/UnB ENE/UnB

Prof. Marcelo Grandi Mandelli
Coordenador do Bacharelado em Ciéncia da Computaçeo

Brasília, 2 de agosto de 2025

Dedicatória

Dedico este trabalho à minha família e amigos.

Agradecimentos

Agradeço a todas as pessoas que dedicaram seu tempo para me auxiliar neste trabalho.

Resumo

O Low Complexity Enhancement Video Coding (LCEVC) atua como uma camada de aprimoramento sobre codificadores convencionais, como o Advanced Video Coding / H.264 (AVC) e o Versatile Video Coding / H.266 (VVC), com foco em eficiência e baixa complexidade computacional. Este trabalho realiza uma análise de qualidade do padrão LCEVC, com o objetivo de identificar os melhores cenários para o LCEVC, onde ele é capaz de superar a qualidade ou a taxa de bits de vídeos que utilizam somente os codificadores que ele aprimora. Estes resultados são analisados para descobrir quais seriam os parâmetros que mais têm a chance de gerar uma codificação eficiente para o LCEVC. Foram conduzidos experimentos variando os parâmetros de quantização da camada base (QP) e da camada de aprimoramento (SW_2), utilizando métricas como Peak Signal-to-Noise Ratio (PSNR) e taxa de bits. Os resultados, analisados com base na Fronteira de Pareto, indicam que o LCEVC apresenta desempenho superior em diversos cenários, principalmente em vídeos com resoluções *Full HD* e *4K*. Além disso, foram identificadas combinações de QP e SW_2 que resultam em codificações mais eficientes, contribuindo para configurações ideais de uso do LCEVC em aplicações de vídeo de alta qualidade com menor custo computacional.

Palavras-chave: LCEVC, AVC, VVC, compressão de vídeo, PSNR, análise de qualidade, TV Digital Brasileira

Abstract

Low Complexity Enhancement Video Coding (LCEVC) operates as an enhancement layer over conventional video encoders, such as Advanced Video Coding / H.264 (AVC) and Versatile Video Coding / H.266 (VVC), focusing on efficiency and low computational complexity. This work presents a quality analysis of the LCEVC standard, aiming to identify the most favorable scenarios where LCEVC can surpass the quality or bitrate of videos encoded solely with the base codecs it enhances. The study also seeks to discover the best parameter configurations for an efficient LCEVC usage. Experiments were conducted by varying the quantization parameters of the base layer (QP) and the enhancement layer ($SW2$), using metrics such as Peak Signal-to-Noise Ratio (PSNR) and bitrate. The results, analyzed through the Pareto Frontier, indicate that LCEVC performs better in several scenarios, especially for *Full HD* and 4K resolution videos. Moreover, specific QP and $SW2$ combinations were found to yield more efficient encodings, contributing to optimal usage configurations of LCEVC for high-quality video applications with lower computational costs.

Keywords: LCEVC, AVC, VVC, video compression, PSNR, quality analysis, Brazilian Digital TV

Sumário

1	Introdução	1
1.1	Motivação	1
1.2	Objetivo	1
1.3	Codificadores de vídeo	2
1.4	Advanced Video Coding / H.264 (AVC)	3
1.5	Versatile Video Coding / H.266 (VVC)	3
1.6	Taxa de bits e Peak Signal-to-Noise Ratio (PSNR)	4
1.6.1	Taxa de bits	4
1.6.2	Peak Signal-to-Noise Ratio (PSNR)	4
1.7	Low Complexity Enhancement Video Coding	5
1.7.1	Codificação	5
1.7.2	Decodificação	6
2	Metodologia	8
2.1	Codificando Vídeos	8
2.2	Análise dos Vídeos	8
2.3	Fronteira de Pareto	9
2.4	Sequências	10
2.5	Programas Usados	11
3	Resultados	12
3.1	AVC	13
3.1.1	Bosphorus	13
3.1.2	ReadySteadyGo	14
3.1.3	Jockey	15
3.1.4	SOCCER	16
3.1.5	City	17
3.1.6	vc-globo-05	18
3.1.7	vc-lcevc-01	19

3.1.8	vc-phillips-01	20
3.1.9	vc-phillips-03	22
3.2	VVC	24
3.2.1	Bosphorus	24
3.2.2	SOCCKER	25
3.2.3	Jockey	26
3.2.4	City	27
3.2.5	vc-phillips-01	28
3.2.6	vc-lcevc-01	30
3.2.7	vc-globo-05	31
3.3	Visão geral dos resultados	32
4	Conclusão	35
4.1	Trabalhos Futuros	36
Referências		37
Anexo		38
I	Scripts	39
I.1	AVC	39
I.2	VVC	49
I.3	Resample	60

Listas de Figuras

1.1	Processo de codificação do LCEVC.	5
1.2	Processo de decodificação do LCEVC.	7
2.1	Exemplo da Fronteira de Pareto. [1]	10
2.2	Frames representativos das sequências de vídeo utilizadas.	11
3.1	Resultados para "Bosphorus" em AVC. [2]	13
3.2	Resultados para "ReadySteadyGo" em AVC. [2]	14
3.3	Resultados para "Jockey" em AVC. [2]	15
3.4	Resultados para "SOCCER" em AVC. [3]	16
3.5	Resultados para "City" em AVC. [3]	17
3.6	Resultados para "vc-globo-05" em AVC. [4]	18
3.7	Resultados para "vc-lcevc-01" em AVC.	19
3.8	Resultados para "vc-philips-01" em AVC.	20
3.9	Resultados para "vc-philips-03"em AVC.	22
3.10	Resultados para "Bosphorus" em VVC. [2]	24
3.11	Resultados para "SOCCER" em VVC. [3]	25
3.12	Resultados para "Jockey" em VVC. [2]	26
3.13	Resultados para "City" em VVC. [3]	27
3.14	Resultados para "vc-philips-01" em VVC.	28
3.15	Resultados para "vc-lcevc-01"em VVC.	30
3.16	Resultados para "vc-globo-05"em VVC [4].	31

Listas de Tabelas

1.1	Interpretação dos valores de PSNR. [5]	5
2.1	Versões dos programas utilizados para codificação e decodificação.	11
3.1	Resultados para Bosphorus em AVC.	13
3.2	Resultados favoráveis do LCEVC para Bosphorus em AVC	14
3.3	Resultados para Jockey em AVC.	15
3.4	Resultados favoráveis do LCEVC para Jockey em AVC.	16
3.5	Resultados para vc-globo-05 em AVC.	18
3.6	Resultados favoráveis do LCEVC para vc-globo-05 em AVC.	19
3.7	Resultados para vc-lcevc-01 em AVC.	20
3.8	Resultados favoráveis do LCEVC para vc-lcevc-01 em AVC.	20
3.9	Resultados para vc-philips-01 em AVC.	21
3.10	Resultados favoráveis do LCEVC para vc-philips-01 em AVC.	21
3.11	Resultados para vc-philips-03 em AVC.	22
3.12	Resultados favoráveis do LCEVC para vc-philips-03 em AVC.	23
3.13	Resultados para Bosphorus em VVC.	24
3.14	Resultados favoráveis do LCEVC para Bosphorus em VVC.	25
3.15	Resultados para Jockey em VVC	26
3.16	Resultados favoráveis do LCEVC para Jockey em VVC.	27
3.17	Resultados para vc-philips-01 em VVC.	28
3.18	Resultados favoráveis do LCEVC para vc-philips-01 em VVC.	29
3.19	Resultados para vc-globo-05 em VVC.	31
3.20	Resultados favoráveis do LCEVC para vc-globo-05 em VVC.	32
3.21	Quantidade de pontos acima da curva de eficiência para o AVC.	32
3.22	Quantidade de pontos acima da curva de eficiência para o VVC.	32
3.23	Quantidade de ocorrências de cada valor de SW2 nos resultados favoráveis ao LCEVC	33
3.24	Frequência de ocorrência dos pares SW2 e QP para o LCEVC nos pontos favoráveis.	34

Lista de Abreviaturas e Siglas

AVC Advanced Video Coding / H.264.

DLSS Deep Learning Super Sampling.

EVC Essential Video Coding.

HEVC High Efficiency Video Coding / H.265.

ITU Telecommunication Standardization Sector.

JVET Joint Video Experts Team.

LCEVC Low Complexity Enhancement Video Coding.

MPEG Moving Picture Experts Group.

PSNR Peak Signal-to-Noise Ratio.

QP Quantization Parameter.

VVC Versatile Video Coding / H.266.

Capítulo 1

Introdução

1.1 Motivação

O crescimento do consumo de vídeo digital, impulsionado por serviços de streaming, transmissões ao vivo e a popularização de dispositivos conectados, trouxe desafios significativos para a transmissão de conteúdos em alta qualidade sem sobrecarregar as redes de comunicação. No contexto brasileiro, a demanda por vídeos em resoluções cada vez maiores, como 4K e 8K, exige soluções de compressão mais eficientes para garantir a entrega de conteúdo com qualidade e baixo custo operacional.

Nesse cenário, o Low Complexity Enhancement Video Coding (LCEVC) surge como uma tecnologia inovadora, capaz de aprimorar a eficiência de codificadores tradicionais ao adicionar camadas de aprimoramento com baixa complexidade computacional. Sua adoção tem ganhado destaque internacionalmente e, mais recentemente, no Brasil, onde foi escolhida como parte do padrão para a próximas gerações da TV Digital, parcialmente na TV 2.5 e por completo na TV 3.0 [6, 7], onde atualmente é usado o codec Advanced Video Coding / H.264 (AVC). Essa escolha reflete o potencial do LCEVC em viabilizar transmissões de alta qualidade com menor uso de banda e recursos computacionais, tornando-se um tema relevante para pesquisa e análise no contexto nacional [8, 9].

1.2 Objetivo

Este trabalho tem como objetivo realizar uma análise qualitativa do desempenho do padrão Low Complexity Enhancement Video Coding (LCEVC), comparando seus resultados com os obtidos por codecs convencionais utilizados de forma isolada, como o AVC e o VVC. Com isso, encontrar casos onde o LCEVC consegue uma vantagem em relação à forma isolada, para tentar encontrar a melhor maneira de utilizar e configurar o LCEVC.

Para isso, são realizadas codificações das mesmas sequências de vídeo sob diferentes configurações de parâmetros, variando o fator de quantização da camada base e os valores de qualidade da camada de aprimoramento. Com base na relação entre PSNR e taxa de bits, busca-se identificar em quais cenários o LCEVC apresenta ganhos reais de qualidade e eficiência.

1.3 Codificadores de vídeo

Quando se cria um vídeo para ser utilizado como mídia em alguma plataforma, é esperado que ele seja recebido e consumido com a melhor qualidade possível. Isso é um desafio que estas plataformas enfrentam há muitos anos. Por exemplo, quando se grava um filme, as filmagens são geradas em um formato bruto que ainda não sofreu nenhum tipo de edição ou compressão. Este formato gera vídeos de tamanhos gigantescos com a melhor qualidade possível [10]. Quando um filme está pronto para ser distribuído, é preciso fazer com que ele caiba em uma mídia física, como um DVD ou Blu-Ray, ou tenha um tamanho aceitável para ser transferido via streaming. Para que isso pudesse ser realizado, criou-se o padrão MPEG-2, que é capaz de codificar vídeos a taxas aproximadas de 4 a 9 Mb/s [11]. Esses algoritmos visam diminuir o tamanho de um vídeo, mas manter a melhor qualidade possível.

Com o aumento significativo do tráfego de dados em redes digitais e a popularização de serviços como streaming e videoconferências, a compressão eficiente de vídeo se tornou um fator crucial para garantir a qualidade destes serviços ao usuário final e também sua viabilidade técnica e econômica para os provedores destes serviços.

Apesar dos avanços destes *codecs*, os ganhos de compressão vieram acompanhados do aumento significativo da complexidade computacional, o que dificulta o seu uso em dispositivos com recursos limitados, como *smartphones*, receptores de TV ou plataformas web.

Segundo a Globo [12], serviços e aplicações envolvendo o uso de sinais de vídeo mantêm uma tendência de crescimento ao longo dos anos, onde o tráfego de vídeo responde por 80% do tráfego presente na Internet. Durante a pandemia, esse cenário incrementou sensivelmente, a ponto de alguns Governos Nacionais cogitarem a redução da oferta de sinais de vídeo de alguns serviços de streaming receando um possível colapso na infraestrutura de internet. Entende-se que a existência de codificadores mais eficientes é mais do que nunca uma forte demanda do mercado, oferecendo excelentes oportunidades para pesquisa e inovação.

Esses algoritmos foram sendo aprimorados com o passar do tempo e novos algoritmos foram sendo criados. Existem inúmeros algoritmos para compressão de áudio e vídeo,

onde o foco agora serão os padrões Advanced Video Coding / H.264 (AVC) e Versatile Video Coding / H.266 (VVC), que serão utilizados neste trabalho.

1.4 Advanced Video Coding / H.264 (AVC)

O AVC é um padrão para compressão de vídeo baseado no MPEG-4 Parte 10, desenvolvido pela Telecommunication Standardization Sector (ITU) em conjunto com a MPEG. É um dos codificadores mais utilizados no mundo, devido a sua alta eficiência de compressão e boa qualidade de imagem, além da ampla compatibilidade.

O AVC é um codificador híbrido baseado em blocos, ou seja, ele divide o frame de um vídeo em pequenos blocos, onde nesses blocos são aplicadas as técnicas de compressão. Isso permite que o processamento seja segmentado facilitando a previsão, transformação e codificação. Ele emprega técnicas de compressão de movimento, transformada discreta do cosseno, quantização escalável e codificação de entropia. Ele possui um modelo de compressão temporal e compressão espacial. Estas características permitem a redução da taxa de bits sem degradação perceptível. Ele define uma arquitetura modular com níveis e perfis, para permitir sua aplicação em diferentes contextos, como streaming ou conteúdo de alta definição.

Este codificador se destaca em relação aos seus predecessores, como o MPEG-2, em eficiência de compressão. Essa característica foi essencial para que ele se tornasse um dos codificadores mais utilizados. [13]

1.5 Versatile Video Coding / H.266 (VVC)

O VVC também é um padrão de codificação de vídeo desenvolvido pela Joint Video Experts Team (JVET), que é uma colaboração entre o Telecommunication Standardization Sector (ITU) e o Moving Picture Experts Group (MPEG). Seu objetivo era dobrar a eficiência de compressão do codificador HEVC, mas mantendo a mesma qualidade visual. O VVC também é um codificador híbrido baseado em blocos, porém com diversas inovações que o tornam mais eficiente. Entre suas principais técnicas estão a unidade quadricomposta com partições adaptativas, melhorias no *intra-prediction*, maior sofisticação na compensação de movimento e filtros de interpolação, introdução de novos modos de transformadas e aprimoramento nas técnicas de codificação entropia.

Este codificador possui um amplo uso como vídeos com resoluções altas (como 8k), com realidade virtual, streaming e vídeo conferências em redes móveis. Possui suporte para HDR e um amplo espaço de cores. Porém, este alto desempenho vem como uma alta complexidade computacional, ainda maior que outros codificadores, especialmente na

codificação, o que demanda otimizações específicas e o uso de aceleração por hardware, ou seja, o uso de placas de vídeo. [14]

1.6 Taxa de bits e Peak Signal-to-Noise Ratio (PSNR)

Para que fosse possível estudar, analisar e comprar se sequências geradas, é necessário ter alguma métrica para poder compará-los. Para este caso, foi escolhido gerar um gráfico relacionando a taxa de bits e o PSNR de cada vídeo gerado dentro de um determinado parâmetro em comum.

1.6.1 Taxa de bits

A taxa de bits pode ser definida como o número de *bits* que é transferido ou processado por alguma unidade de tempo [15]. Este valor dá uma ideia da quantidade de recursos que serão necessários para transmitir e decodificar cada vídeo. Quanto maior este valor, maior foi o tempo necessário para a codificação, maior é o tamanho do arquivo, que acarreta em uma quantidade maior de dados sendo transferidos, caso for utilizado para streaming, e maior é a computação necessária para que o vídeo seja reconstruído pelo de codificador. Então, este valor nos ajuda a determinar se um determinado ponto é ou não viável. Pode-se calcular a taxa de bits com a seguinte fórmula:

$$\text{Taxa de bits} = \frac{\text{Tamanho (em bits)}}{\text{Duração (em segundos)}}$$

1.6.2 Peak Signal-to-Noise Ratio (PSNR)

Signal-to-Noise Ratio, ou Relação Sinal-Ruído, é uma medida que compara a força de um sinal desejado com a força do ruído de fundo. É uma métrica utilizada para avaliar a qualidade e a confiabilidade de um sinal [15]. Para utilizar esta métrica na análise de vídeos, costuma-se usar o PSNR, que é a relação entre a máxima energia de um sinal e o ruído que afeta sua representação. Por muitos sinais terem sua amplitude dinâmica, o PSNR é normalmente expressado por uma escala logarítmica em decibéis.

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{\text{MAX}^2}{\text{MSE}} \right)$$

Onde:

- PSNR: Peak Signal-to-Noise Ratio (em decibéis, dB);
- MAX: valor máximo possível de um pixel (por exemplo, 255 em 8 bits);

- MSE: erro quadrático médio entre os quadros original e comprimido;

Para este estudo, o importante é saber interpretar quanto cada valor representa na qualidade final do vídeo. Para isso, esta tabela demonstra a interpretação básica destes valores:

PSNR > 33 dB	Qualidade Excelente
33 dB > PSNR > 30 dB	Qualidade Aceitável
PSNR < 30 dB	Qualidade Ruim

Tabela 1.1: Interpretação dos valores de PSNR. [5]

1.7 Low Complexity Enhancement Video Coding

O LCEVC é uma tecnologia de codificação em múltiplas camadas desenvolvida para atuar como uma camada de aprimoramento sobre os codificadores já existentes, como os que foram mencionados anteriormente.

O LCEVC, por ser um codificador de vídeo, o LCEVC possui o processo de codificação e decodificação do vídeo, onde o decodificador é responsável por reconstruir o vídeo.

1.7.1 Codificação

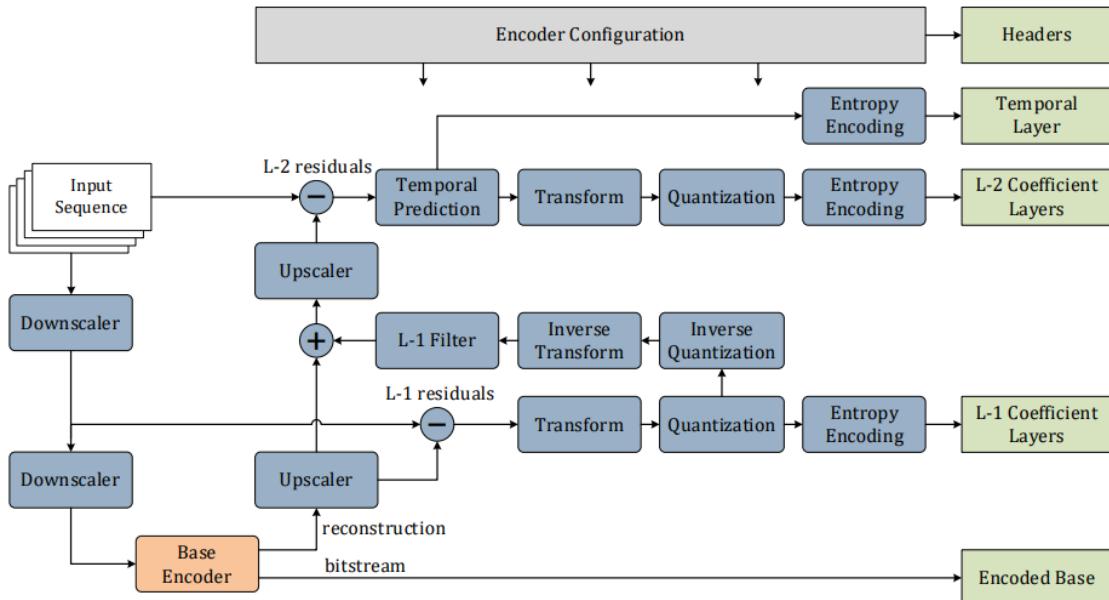


Figura 1.1: Processo de codificação do LCEVC (Fonte: [16]).

Camada Base

Primeiro, a sequência de entrada possui sua resolução reduzida por um processo chamado de *downscaling*. Ela passa por dois processos de *downscaling* que utilizam o parâmetro de modo de escalonamento (*scaling mode*) definido. Com o vídeo reduzido, então, é acionado o codificador responsável por codificar a camada base de acordo com o modo escolhido, que pode ser AVC, VVC, HEVC ou EVC. O bitstream gerado pode ser incluído diretamente no fluxo LCEVC. [16, 17]

Subcamada de Aprimoramento 1 (L-1)

A imagem reconstruída da camada base é reamostrada para a resolução original e subtraída da versão redimensionada da sequência original. Essa diferença forma o *resíduo L-1*. Este é processado por ferramentas de codificação específicas (transformada, quantização e codificação de entropia), gerando coeficientes quantizados codificados. Esses dados compõem a primeira subcamada da camada de aprimoramento. [16, 17]

Subcamada de Aprimoramento 2 (L-2)

A reconstrução da subcamada L-1 é reconstruída internamente e, dependendo do modo de escalonamento, reamostrada novamente. A subtração da sequência de entrada original com essa reconstrução gera o *resíduo L-2*, que também é transformado e quantizado. Nesta etapa, é possível aplicar predição temporal aos coeficientes transformados, aumentando a eficiência de codificação. Os dados gerados, incluindo a informação de predição temporal por bloco, são inseridos no bitstream final. [16, 17]

1.7.2 Decodificação

O processo de decodificação de um vídeo com o LCEVC é composto por três etapas principais: decodificação da camada base, reconstrução da subcamada de aprimoramento 1 (L-1) e aplicação da subcamada de aprimoramento 2 (L-2). Abaixo, descrevemos cada uma dessas etapas.

Camada Base

O decodificador extrai e decodifica o bitstream da camada base utilizando o codec tradicional configurado. A imagem reconstruída, chamada de *Decoded Base Picture*, pode ser reamostrada (upscaleing) conforme o modo de escalonamento utilizado na codificação, formando a *Preliminary Intermediate Picture*. [16, 17]

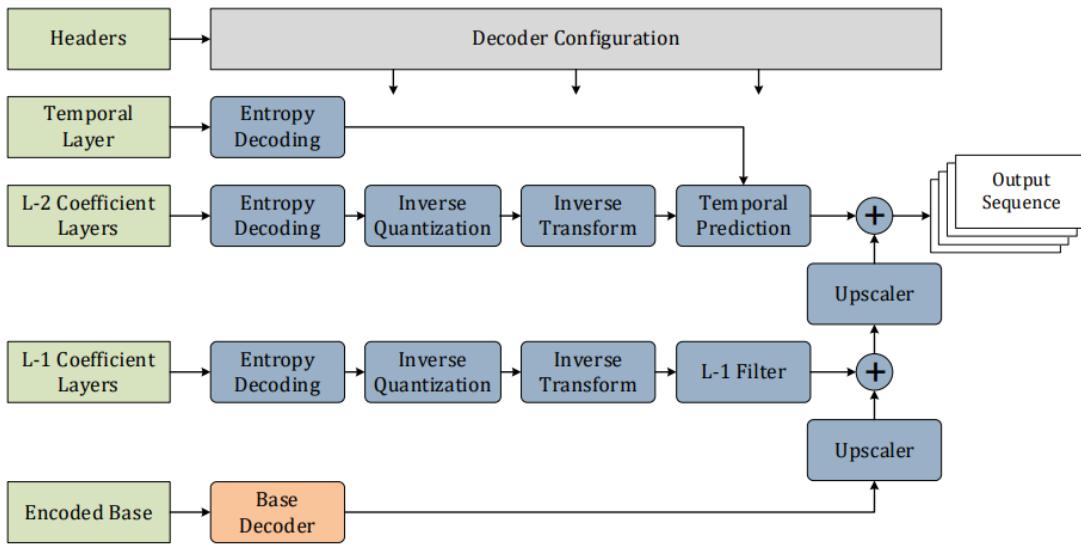


Figura 1.2: Processo de decodificação do LCEVC (Fonte: [16]).

Subcamada de Aprimoramento 1 (L-1)

Os coeficientes quantizados da subcamada [17]L-1 são decodificados utilizando as ferramentas inversas do processo de codificação (decodificação de entropia, desquantização e transformada inversa). Opcionalmente, é aplicado um filtro L-1 para suavizar as bordas dos blocos transformados. O resultado é somado à *Preliminary Intermediate Picture*, formando a *Combined Intermediate Picture*. Em seguida, pode ser aplicada uma nova etapa de upscaling para obter a *Preliminary Output Picture*. [16, 17]

Subcamada de Aprimoramento 2 (L-2)

Por fim, a segunda subcamada de aprimoramento é decodificada. Caso os metadados indiquem o uso de predição temporal, os coeficientes são ajustados com base nos resíduos temporais do quadro anterior. Após a decodificação dos coeficientes, o resultado da subcamada L-2 é somado à *Preliminary Output Picture*, produzindo o quadro final decodificado, denominado *Combined Output Picture*. [16, 17]

Capítulo 2

Metodologia

2.1 Codificando Vídeos

Para codificar os vídeos para o LCEVC, foi usado o codificador LTM Model Encoder e Decoder, obtido através do Git da MPEG. Para codificar um vídeo utilizando este codificador, são passados os parâmetros pela linha de comando, ou através de um arquivo de configuração que possui os valores padrões do codificador. Como mencionado anteriormente, o LCEVC é constituído por duas camadas, a camada base contendo um vídeo com a metade da resolução do vídeo original, e uma camada de aprimoramento, que possui as informações para que o LCEVC possa realizar um upscaling no vídeo para que ele fique o mais parecido com o vídeo original possível. A camada base pode estar nos formatos AVC, VVC, HEVC e EVC. Já a camada de aprimoramento, possui duas subcamadas, onde uma é opcional e a outra é obrigatória, contendo os principais dados.

Para este trabalho, foi alterado o parâmetro que determina o nível da qualidade que será destinada para esta subcamada. Este valor determina a quantização desta camada, onde o quanto menor o valor, melhor a qualidade que esta camada possuirá, consequentemente aumentando o tamanho que a camada do LCEVC terá em relação ao arquivo final. Este valor é chamado de $SW2$, e seu parâmetro é `--cq_step_width_loq_0`. Além do valor de $SW2$ que modifica a qualidade da camada do LCEVC, foi modificado o valor do QP para a camada base. Para cada valor de QP , foi usado uma sequência de valores fixos para o $SW2$, para que fosse possível comparar a relação que estes valores iriam influenciar na qualidade e tamanho final do arquivo.

2.2 Análise dos Vídeos

Neste trabalho, será analisado o resultado de várias codificações utilizando o LCEVC, para tentar demonstrar qual seriam os melhores parâmetros para o seu uso, levando em

consideração os seus casos de uso. O programa utilizado foi compilado para Linux e sempre utilizado em uma distribuição Linux com base em Ubuntu.

Foi desenvolvido um *script Bash* para Linux [18] que executa o programa codificador paralelamente para cada valor de QP utilizado. O *script* cria uma execução por QP e processa sequencialmente por valor de $SW2$. Os valores utilizados foram:

QPs: 22, 25, 27, 30, 32, 35, 37.

SW2: 250, 750, 1250, 1750, 2250, 2750, 3250.

Além das codificações utilizando LCEVC, também foram codificados vídeos no mesmo padrão de compressão utilizado pelo LCEVC, mas sem o LCEVC, para que fosse possível ter uma referência da qualidade máxima possível obter sem o LCEVC. Neste caso, foram usados os mesmos valores de QP e utilizado a resolução final que o vídeo do LCEVC teria. Também foram extraídos os parâmetros que o LCEVC altera para o *downsampling* da camada base e adicionados no comando de execução da codificação sem o LCEVC.

Após cada vídeo ser codificado, o *script* também faz o cálculo de Peak Signal-to-Noise Ratio (PSNR) do resultado obtido, convertido para o formato YUV, assim como o arquivo original.

2.3 Fronteira de Pareto

A *Fronteira de Pareto* é uma técnica utilizada para analisar a relação entre dois ou mais objetivos, onde é possível identificar quais soluções são eficientes em relação a outros objetivos [19]. Neste trabalho, a Fronteira de Pareto é utilizada para analisar a relação entre a qualidade do vídeo, medida pelo PSNR, e o tamanho do arquivo, medido pela taxa de bits. A Fronteira de Pareto é representada por um conjunto de pontos que não podem ser melhorados em um objetivo sem piorar outro objetivo. Ou seja, se um ponto está na fronteira de Pareto, não é possível aumentar a qualidade do vídeo sem aumentar o tamanho do arquivo, ou diminuir o tamanho do arquivo sem diminuir a qualidade do vídeo.

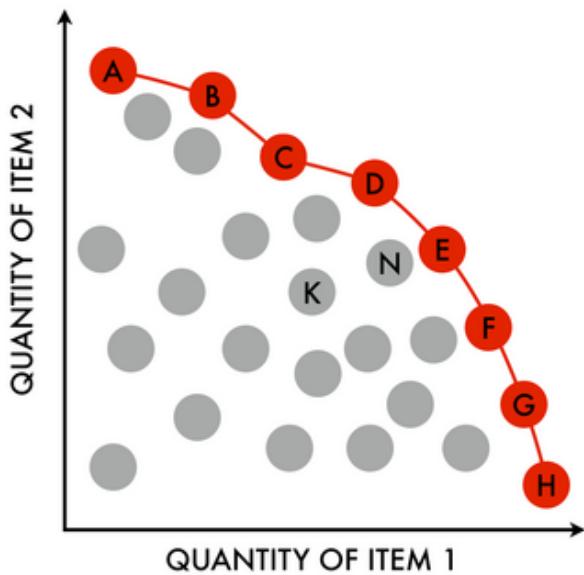


Figura 2.1: Exemplo da Fronteira de Pareto. [1]

2.4 Sequências

As sequências utilizadas para os testes deste trabalho foram obtidas em diversos locais diferentes, como: Ultra Video Group [2], Xiph.org [3] e vídeos que foram utilizados para testes da TV Digital, fornecidos pelo Laboratório de Engenharia Elétrica da UnB. A seguir uma breve descrição do conteúdo de cada sequência.

- Bosphorus: Um barco navegando pelo Estreito de Bósforo;
- Jockey: Um jóquei montado em seu cavalo correndo em um hipódromo. A câmera está com zoom e seguindo o cavalo.
- ReadySteadyGo: Uma largada de corrida de cavalos.
- vc-globo-05: Trecho de uma transmissão de futebol da Seleção Brasileira pela TV Globo.
- vc-lcevc-01: Um vídeo mostrando o horizonte e uma tabela fixa na tela.
- vc-phillips-01: Um campo em um dia claro. Plantas ao vento próximo da câmera.
- vc-phillips-03: Um vídeo de um canal em uma cidade.
- City: Uma visão aérea de Nova Iorque, focando em um prédio.
- SOCCER: Um trecho de um treinamento de futebol.

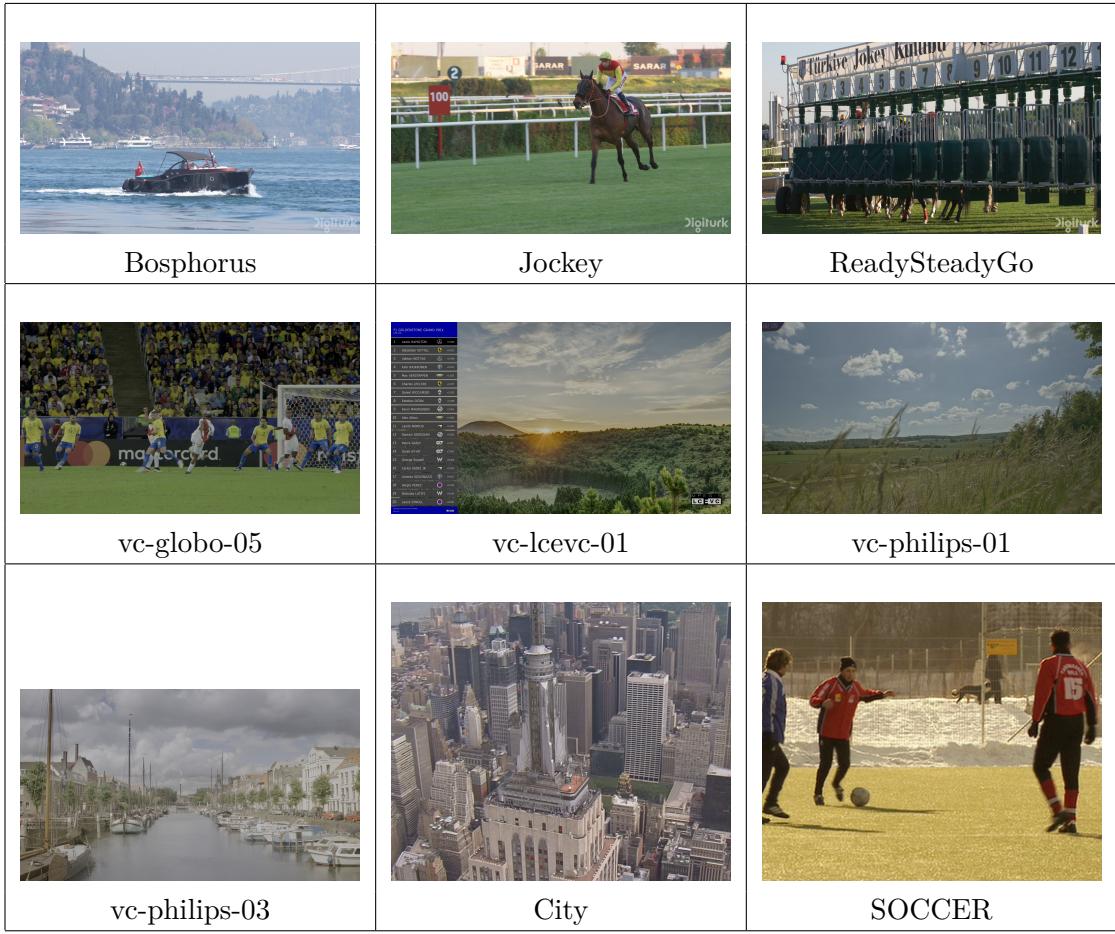


Figura 2.2: Frames representativos das sequências de vídeo utilizadas.

2.5 Programas Usados

Como mencionado previamente, o codificador e decodificador principal é o "*Test Model of Low Complexity Enhancement Video Coding*" LTM [20]. Quando é realizado a codificação do LCEVC com o formato escolhido da base, o LTM chama outro programa para codificar ou decodificar o arquivo. No caso deste trabalho, ao escolher o AVC, o LTM chama o JM, e no caso do VVC, o VTM é acionado. As versões utilizados estão na 2.1.

Programa	Versão
LTM (LCEVC Test Model)	7.0
JM (AVC Reference Software)	19.0
VTM (VVC Test Model)	12.0

Tabela 2.1: Versões dos programas utilizados para codificação e decodificação.

Capítulo 3

Resultados

Após a execução do Script de codificação, os dados obtidos são armazenados em um arquivo CSV. Com os resultados finais, outro script em *Python* é executado, onde ele cria um gráfico com estes dados. Abaixo estão os resultados obtidos.

A análise dos gráficos gerados para as sequências testadas permite observar o comportamento do LCEVC em diferentes condições de codificação. Foram analisados dois codificadores: AVC e VVC, com e sem o uso do LCEVC como camada de aprimoramento.

No geral, os gráficos PSNR x Taxa de bits permitem avaliar a eficiência da compressão, considerando que uma melhor relação é obtida quando se atinge maior qualidade (PSNR) com menor taxa de bits. A seguir, apresenta-se uma análise detalhada por cenário. Para alguns resultados, se um vídeo em LCEVC resultou em uma taxa de bits muito alta, ficando distante dos valores do codificador base, eles foram omitidos para melhor análise e visualização nos gráficos.

Além disso, para que o gráfico pudesse ficar legível, só foi rotulado o valor de SW2 para o parâmetro de QP37.

3.1 AVC

3.1.1 Bosphorus

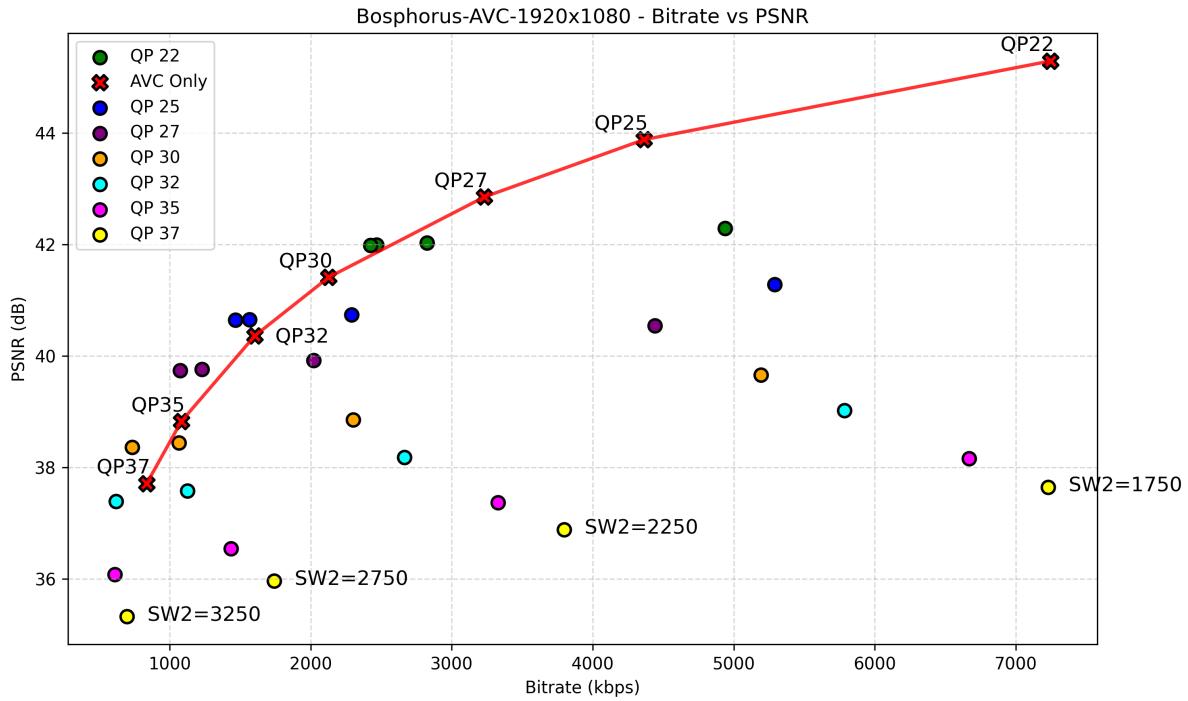


Figura 3.1: Resultados para "Bosphorus" em AVC. [2]

Para a sequência Bosphorus, o LCEVC conseguiu ótimos resultados. Mesmo com uma grande dispersão pela tabela e vários resultados ineficientes, certos valores de SW2 conseguiram atingir resultados positivos.

Para o LCEVC, vemos resultados positivos para QP 30, 27, 25 e 22. Estes QPs conseguiram resultados superiores à curva de eficiência dos vídeos de referência. Os resultados que apresentaram um resultado superior ao AVC estão abaixo.

QP	Tamanho (b)	PSNR (dB)	Taxa (kbps)
37	1047004	37.71	837.60
35	1352315	38.82	1081.85
32	2005088	40.36	1604.07
30	2660081	41.41	2128.06
27	4038641	42.85	3230.91
25	5455551	43.88	4364.44
22	9055951	45.29	7244.76

Tabela 3.1: Resultados para Bosphorus em AVC.

SW2	QP	Tamanho (b)	Proporção	PSNR (dB)	Taxa (kbps)	Superou
3250	30	916195	10.03%	38.36	732.96	AVC QP37
3250	27	1342845	2.68%	39.74	1074.28	AVC QP35
2750	27	1536124	14.93%	39.76	1228.90	AVC QP35
3250	25	1831695	2.30%	40.64	1465.36	AVC QP32
2750	25	1958104	8.61%	40.65	1566.48	AVC QP32
3250	22	3031413	1.21%	41.98	2425.13	AVC QP30-27
2750	22	3083003	2.86%	41.99	2466.40	AVC QP30-27

Tabela 3.2: Resultados favoráveis do LCEVC para Bosphorus em AVC

Estes resultados mostram cenários onde o LCEVC se mostrou superior ao uso de somente AVC e o respectivo resultado que ele superou.

3.1.2 ReadySteadyGo

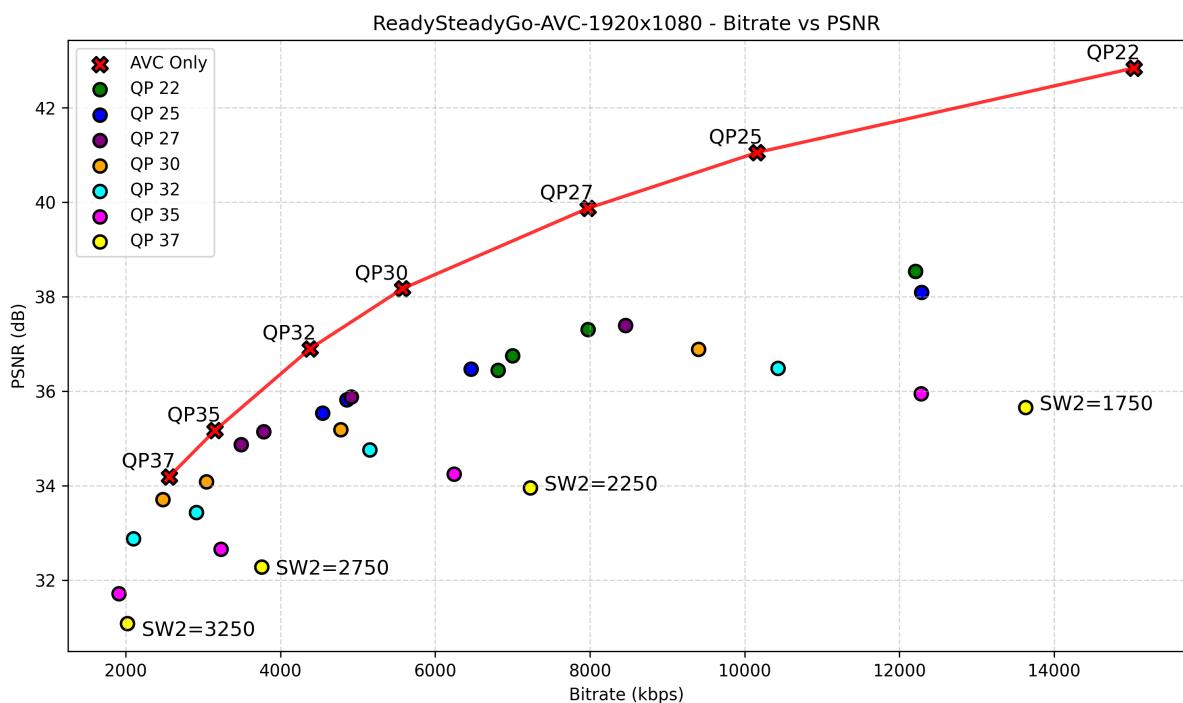


Figura 3.2: Resultados para "ReadySteadyGo" em AVC. [2]

Para esta sequência, o LCEVC não conseguiu atingir resultados satisfatórios. Todos os seus resultados foram abaixo da curva do AVC e não mostrou resultados que justifiquem o seu uso baseado em melhora de qualidade ou taxa. Porém, os resultados para um SW2 maior, ficaram relativamente próximos, como é o caso do QP 30 e 27 do LCEVC. Nestes casos, o uso do LCEVC manteria o PSNR próximo ao resultado obtido para a referência

em QP 37 e 35, onde, por exemplo, o uso do LCEVC para se adequar a um novo padrão não seria perceptível.

3.1.3 Jockey

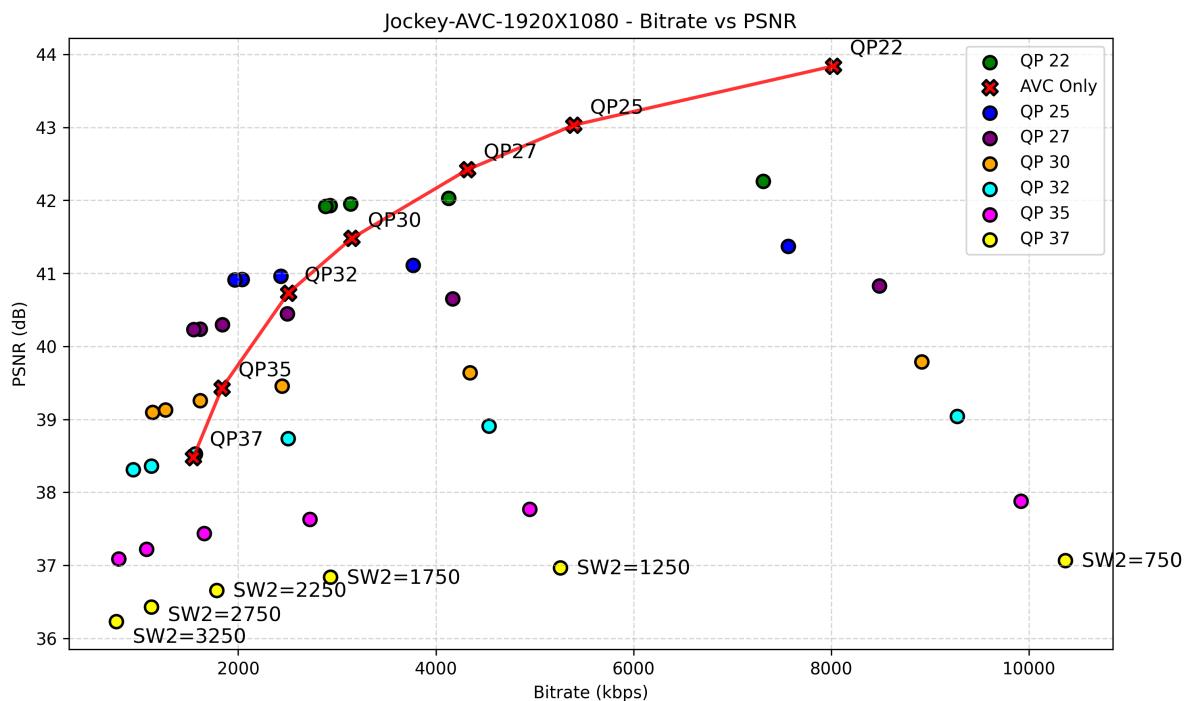


Figura 3.3: Resultados para "Jockey" em AVC. [2]

Aqui é outro caso onde o LCEVC obteve resultados satisfatórios. Vários resultados estão acima da curva do AVC, demonstrando uma ótima performance o LCEVC para os parâmetros escolhidos. Os valores de QP para LCEVC que se destacaram foram 32, 30, 27 e 25, que ficaram acima da curva de referência. Agora, levando em consideração somente os pontos que estão acima dos pontos do AVC, temos o seguinte resultado.

QP	Tamanho (b)	PSNR (dB)	Taxa (kbps)
37	1932479	38.48	1545.98
35	2294827	39.43	1835.86
32	3132841	40.73	2506.27
30	3935452	41.48	3148.36
27	5398380	42.42	4318.70
25	6743403	43.03	5394.72
22	10029224	43.84	8023.38

Tabela 3.3: Resultados para Jockey em AVC.

SW2	QP	Tamanho (b)	Proporção	PSNR (dB)	Taxa (kbps)	Superou
2250	32	1955243	45.60%	38.53	1564.19	AVC QP37
3250	30	1417144	4.45%	39.10	1133.72	AVC QP37
2750	30	1580745	14.34%	39.13	1264.60	AVC QP37
2250	30	2019083	32.94%	39.26	1615.27	AVC QP37
3250	27	1934343	1.69%	40.23	1547.47	AVC QP35
2750	27	2017641	5.75%	40.24	1614.11	AVC QP35
2250	27	2298071	17.25%	40.30	1838.46	AVC QP35
3250	25	2455423	1.78%	40.91	1964.34	AVC QP32
2750	25	2549768	5.41%	40.92	2039.81	AVC QP32
2250	25	3036843	20.58%	40.96	2429.47	AVC QP32
3250	22	3603619	1.11%	41.92	2882.90	AVC QP30
2750	22	3658403	2.59%	41.93	2926.72	AVC QP30
2250	22	3923132	9.16%	41.95	3138.51	AVC QP30

Tabela 3.4: Resultados favoráveis do LCEVC para Jockey em AVC.

3.1.4 SOCCER

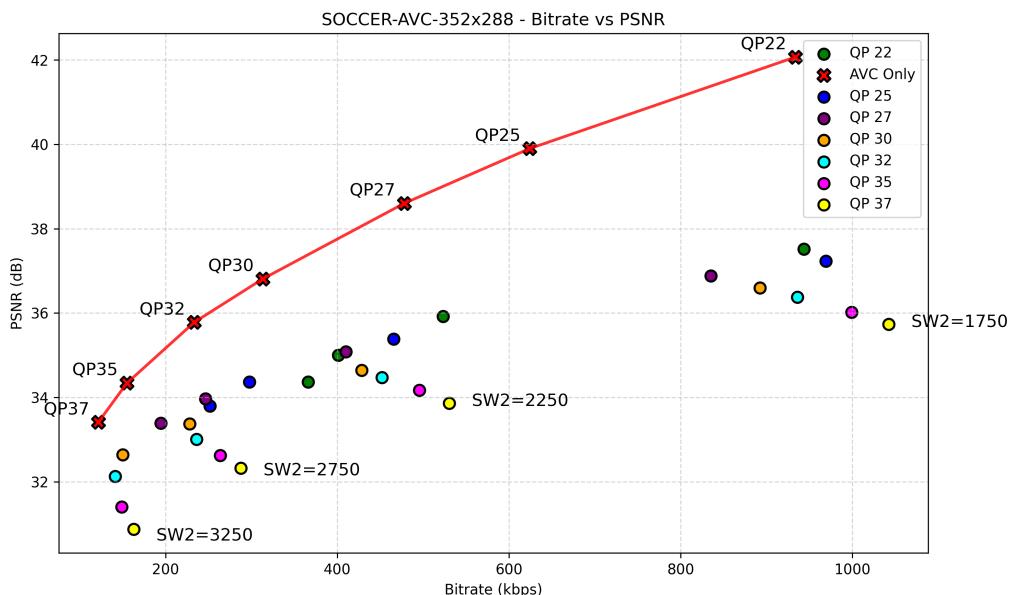


Figura 3.4: Resultados para "SOCCER" em AVC. [3]

Os resultados indicam que o LCEVC não apresentou vantagem clara em relação a somente o AVC na maioria das configurações testadas. Para esta sequência com uma resolução menor, ele não conseguiu atingir resultados satisfatórios. Assim, não houve benefícios em utilizar o LCEVC com AVC nesta sequência, pois os resultados usando somente o AVC já obtiveram o melhor resultado.

3.1.5 City

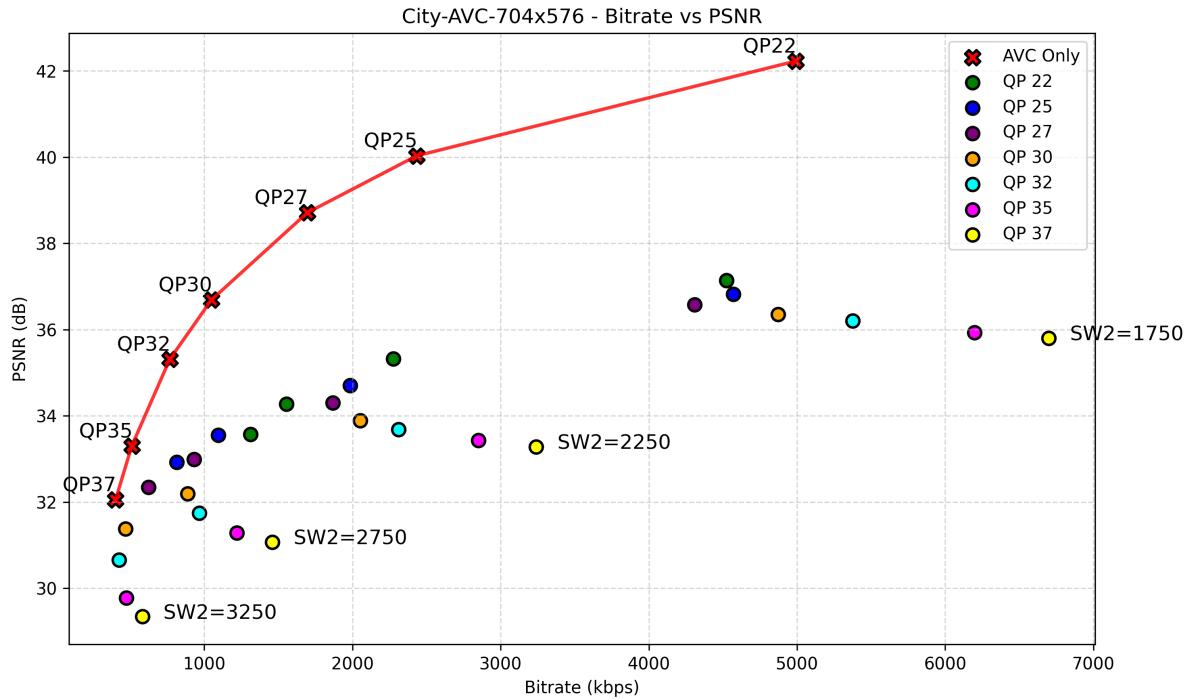


Figura 3.5: Resultados para "City" em AVC. [3]

Na sequência City, o LCEVC não demonstrou uma performance satisfatória. Mais uma sequência com uma resolução menor que 1080p que o LCEVC não obtém um resultado ótimo. Neste caso, o LCEVC não demonstrou um benefício em seu uso, no caso da relação qualidade-tamanho, e não obteve um PSNR superior à curva de referência em nenhum ponto.

3.1.6 vc-globo-05

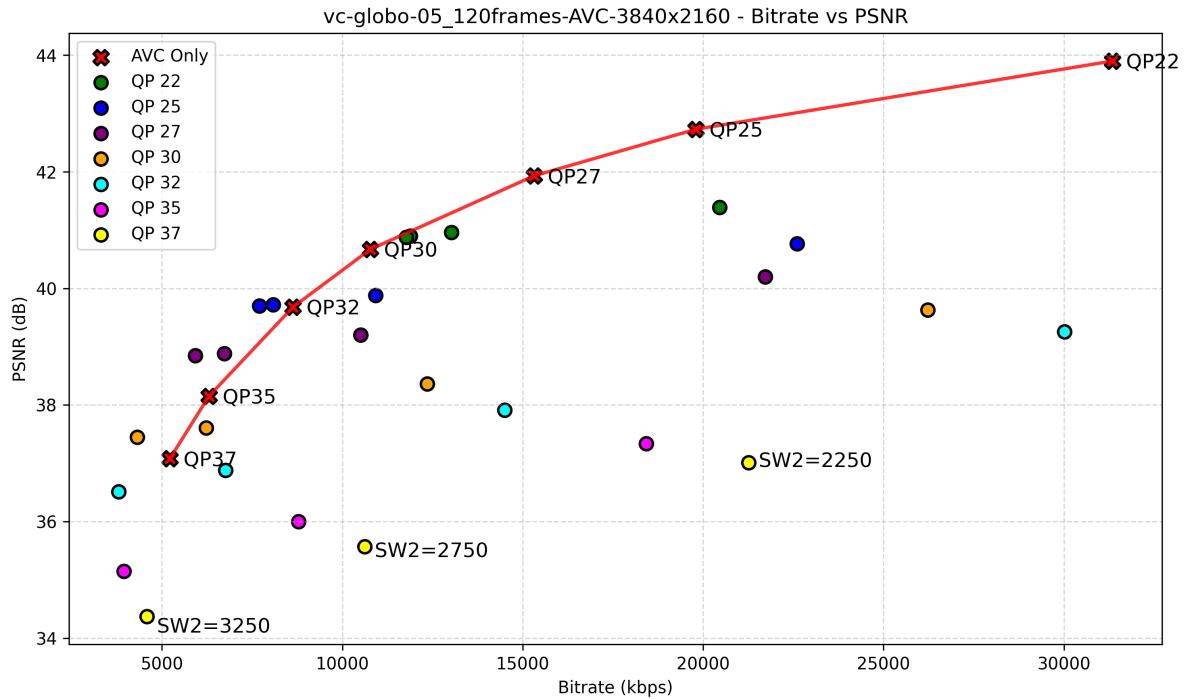


Figura 3.6: Resultados para "vc-globo-05" em AVC. [4]

Para esta sequência, o LCEVC conseguiu obter resultados satisfatórios acima da curva de referência do AVC. Os QPs que se destacaram para o LCEVC foram: 30, 27 e 25. Os resultados obtidos para QP 22 foram muito próximos da curva, onde mesmo não estando acima, ainda representa um valor satisfatório, por manter a qualidade que o AVC obteve. Os resultados favoráveis ao LCEVC estão abaixo.

QP	Tamanho (b)	PSNR (dB)	Taxa (kbps)
37	1305057	37.08	5220.23
35	1575423	38.15	6301.69
32	2158588	39.68	8634.35
30	2694302	40.67	10777.21
27	3828463	41.93	15313.85
25	4950018	42.73	19800.07
22	7836829	43.90	31347.32

Tabela 3.5: Resultados para vc-globo-05 em AVC.

SW2	QP	Tamanho (b)	Proporção	PSNR (dB)	Taxa (kbps)	Superou
3250	30	1079936	8.22%	37.45	4319.74	AVC QP37
3250	27	1480606	1.49%	38.85	5922.42	AVC QP35
2750	27	1681499	13.26%	38.88	6726.00	AVC QP35
3250	25	1924885	0.68%	39.70	7699.54	AVC QP32
2750	25	2021470	5.43%	39.72	8085.88	AVC QP32

Tabela 3.6: Resultados favoráveis do LCEVC para vc-globo-05 em AVC.

3.1.7 vc-lcevc-01

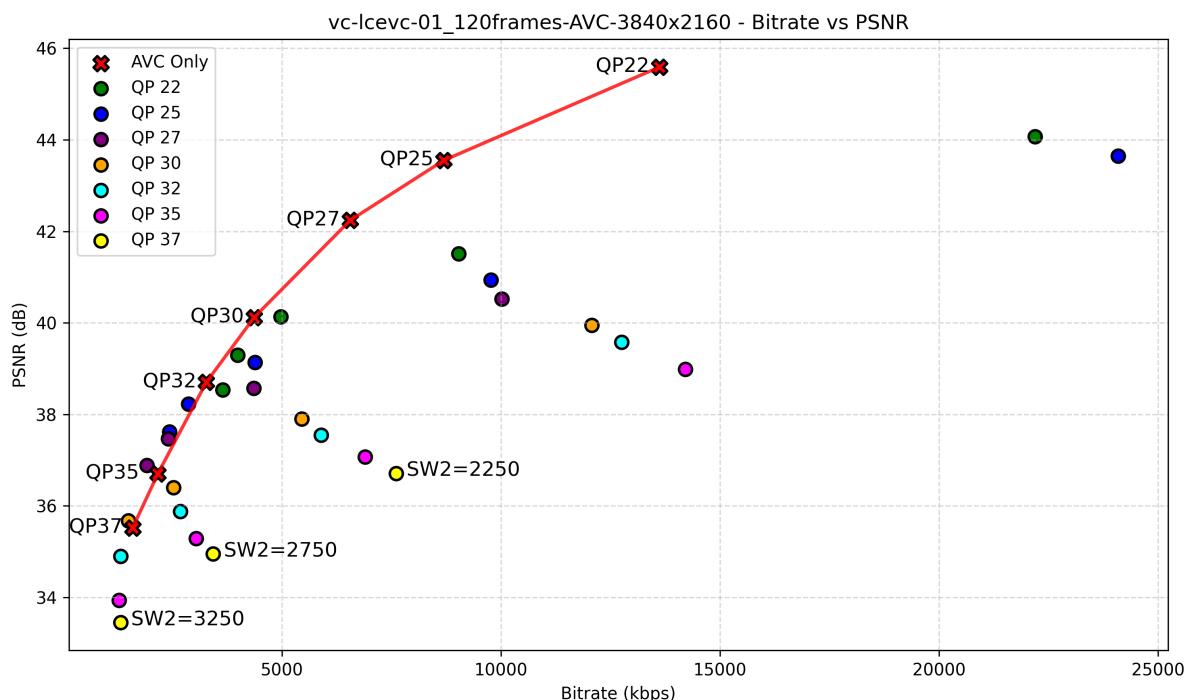


Figura 3.7: Resultados para "vc-lcevc-01" em AVC.

Nesta sequência, o LCEVC conseguiu resultados relevantes para QP igual a 30, 27 e 25. Para o QP de 22, ele não conseguiu alcançar a curva do AVC. Os resultados favoráveis ficaram bem próximos da curva do AVC. Os resultados são apresentados a seguir.

QP	Tamanho (b)	PSNR (dB)	Taxa (kbps)
37	398114	35.52	1592.46
35	541605	36.70	2166.42
32	818304	38.70	3273.22
30	1091870	40.12	4367.48
27	1639723	42.24	6558.89
25	2173840	43.55	8695.36
22	3405589	45.59	13622.36

Tabela 3.7: Resultados para vc-lcevc-01 em AVC.

SW2	QP	Tamanho (b)	Proporção	PSNR (dB)	Taxa (kbps)	Superou
3250	30	373846	28.13%	35.68	1495.38	AVC QP37
3250	27	480407	17.20%	36.89	1921.63	AVC QP35
2750	27	600914	33.80%	37.47	2403.66	AVC QP35
3250	25	608435	13.11%	37.62	2433.74	AVC QP35
2750	25	715498	26.11%	38.23	2861.99	AVC QP35

Tabela 3.8: Resultados favoráveis do LCEVC para vc-lcevc-01 em AVC.

3.1.8 vc-phillips-01

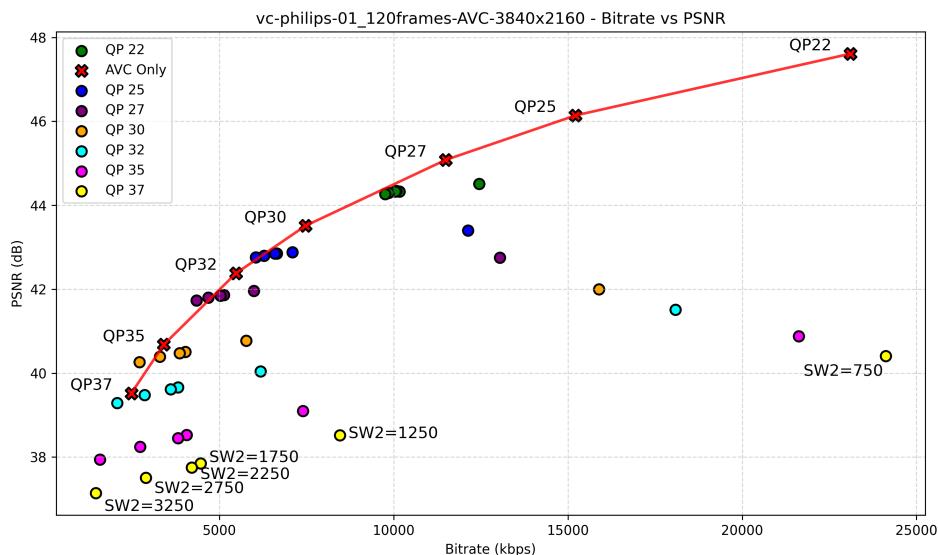


Figura 3.8: Resultados para "vc-phillips-01" em AVC.

Aqui, o LCEVC obteve alguns resultados acima da curva do AVC, mostrando obter resultados superiores ao AVC puro. Os QPs que se destacaram neste resultado foram 30, 27, e 25. Estes valores conseguiram resultados acima da curva.

QP	Tamanho (b)	PSNR (dB)	Taxa (kbps)
37	617900	39.52	2471.60
35	850687	40.68	3402.75
32	1368899	42.38	5475.60
30	1866486	43.51	7465.94
27	2872324	45.08	11489.30
25	3803303	46.14	15213.21
22	5775373	47.61	23101.49

Tabela 3.9: Resultados para vc-philips-01 em AVC.

SW2	QP	Tamanho (b)	Proporção	PSNR (dB)	Taxa (kbps)	Superou
3250	30	54458	8.05%	40.26	2704.88	AVC QP37
3250	27	20276	1.87%	41.73	4338.48	AVC QP37
2750	27	104990	8.98%	41.80	4677.34	AVC QP35
3250	25	9350	0.62%	42.76	6043.10	AVC QP32-30

Tabela 3.10: Resultados favoráveis do LCEVC para vc-philips-01 em AVC.

3.1.9 vc-phillips-03

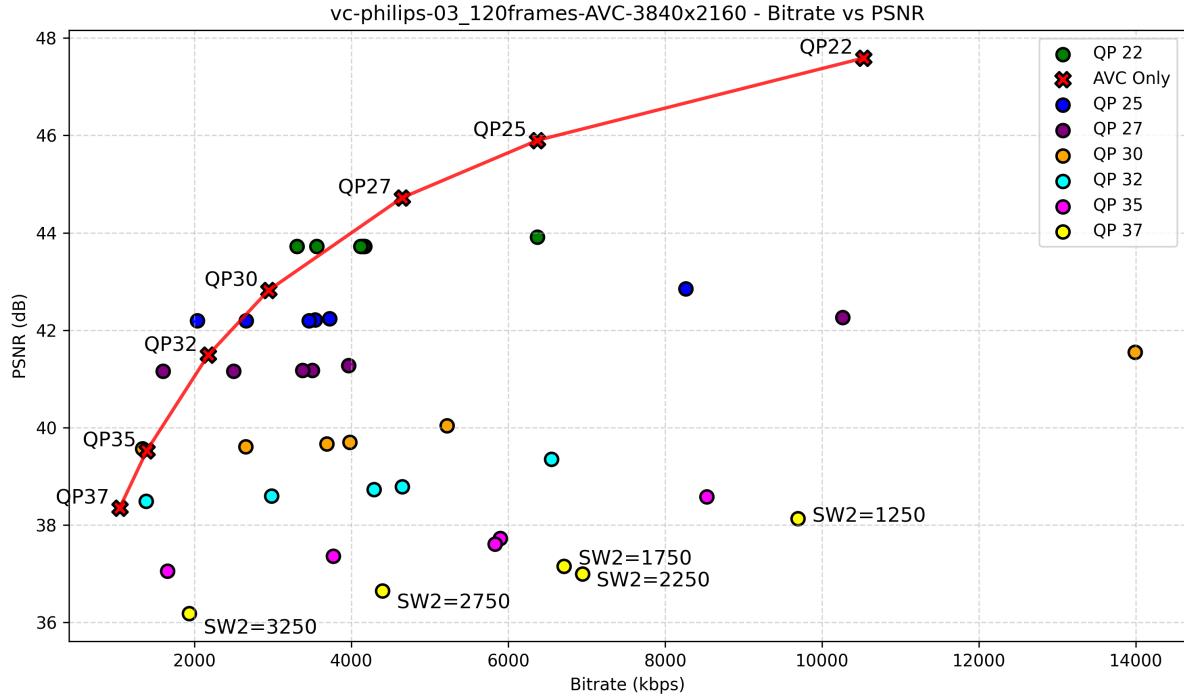


Figura 3.9: Resultados para "vc-phillips-03" em AVC.

Para esta sequência, o LCEVC gerou resultados melhores que o AVC puro. Novamente, os QPs menores se saíram melhor, como esperado, com destaque para os valores de QP de 30, 27, 25 e 22.

QP	Tamanho (b)	PSNR (dB)	Taxa (kbps)
37	261857	38.35	1047.43
35	348522	39.53	1394.09
32	543963	41.49	2175.85
30	736459	42.82	2945.84
27	1161656	44.72	4646.62
25	1593342	45.90	6373.37
22	2633079	47.59	10532.32

Tabela 3.11: Resultados para vc-phillips-03 em AVC.

SW2	QP	Tamanho (b)	Proporção	PSNR (dB)	Taxa (kbps)	Superou
3250	30	334155	31.98%	39.57	1336.62	AVC QP35
3250	27	399297	9.08%	41.16	1597.19	AVC QP35
3250	25	509493	2.68%	42.20	2037.97	AVC QP32
3250	22	826826	0.61%	43.72	3307.30	AVC QP30
2750	22	890314	7.70%	43.72	3561.26	AVC QP30

Tabela 3.12: Resultados favoráveis do LCEVC para vc-phillips-03 em AVC.

3.2 VVC

3.2.1 Bosphorus

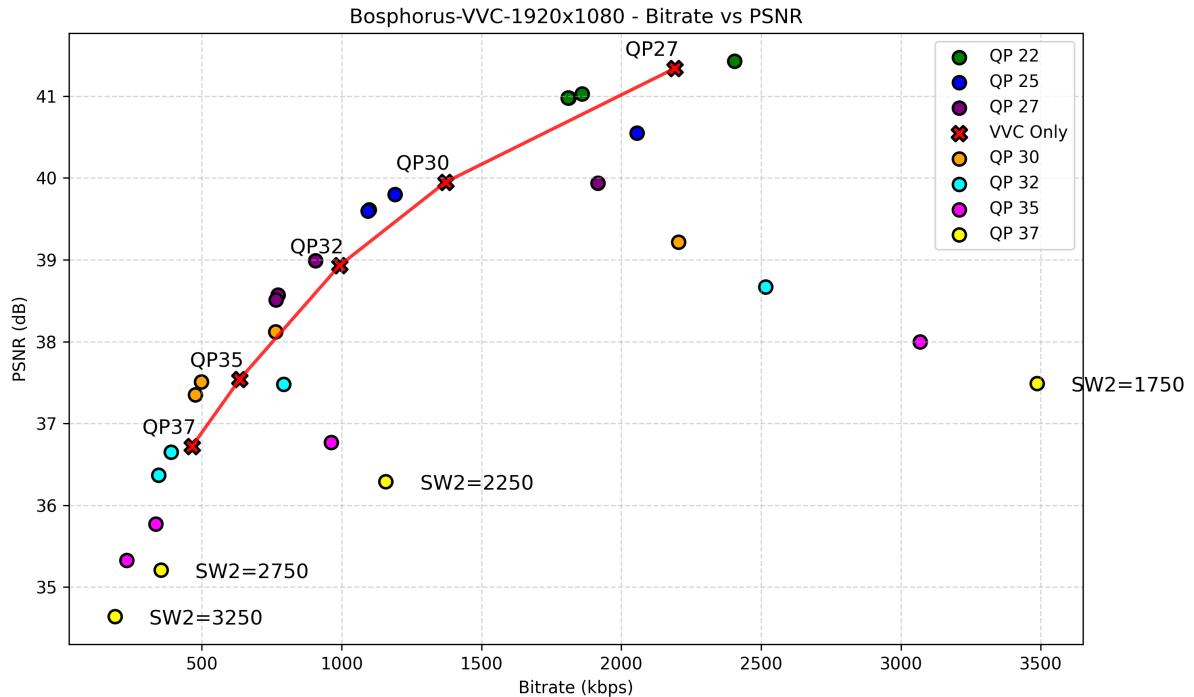


Figura 3.10: Resultados para "Bosphorus" em VVC. [2]

Para o Bosphorus em VVC, o LCEVC também obteve bons resultados. Vários pontos se encontram acima da curva dos vídeos somente em VVC. Os valores para o QP que alcançaram estes pontos foram 30, 27, 25 e 22. Observa-se que se a curva dos vídeos em VVC mantivessem a mesma tendência, haveriam mais pontos favoráveis para o LCEVC.

QP	Tamanho (b)	PSNR (dB)	Taxa (kbps)
37	580476	36.72	464.38
35	793851	37.54	635.08
32	1239951	38.93	991.96
30	1714361	39.95	1371.49
27	2738770	41.34	2191.02

Tabela 3.13: Resultados para Bosphorus em VVC.

SW2	QP	Tamanho (b)	Proporção	PSNR (dB)	Taxa (kbps)	Superou
3250	30	594855	3.25%	37.35	475.88	VVC QP37
2250	30	954971	39.74%	38.12	763.98	VVC QP37
2750	30	621725	7.43%	37.51	497.38	VVC QP35-32
3250	27	955429	1.73%	38.51	764.34	VVC QP35-32
2750	27	965637	2.77%	38.57	772.51	VVC QP35-32
2250	27	1132188	17.07%	38.99	905.75	VVC QP35-32
2750	25	1373014	2.61%	39.61	1098.41	VVC QP32-30
3250	25	1367475	2.22%	39.60	1093.98	VVC QP32-30
2250	25	1487470	10.11%	39.80	1189.98	VVC QP32-30
2750	22	2264797	1.48%	40.98	1811.84	VVC QP30-27
3250	22	2261661	1.34%	40.98	1809.33	VVC QP30-27

Tabela 3.14: Resultados favoráveis do LCEVC para Bosphorus em VVC.

3.2.2 SOCCER

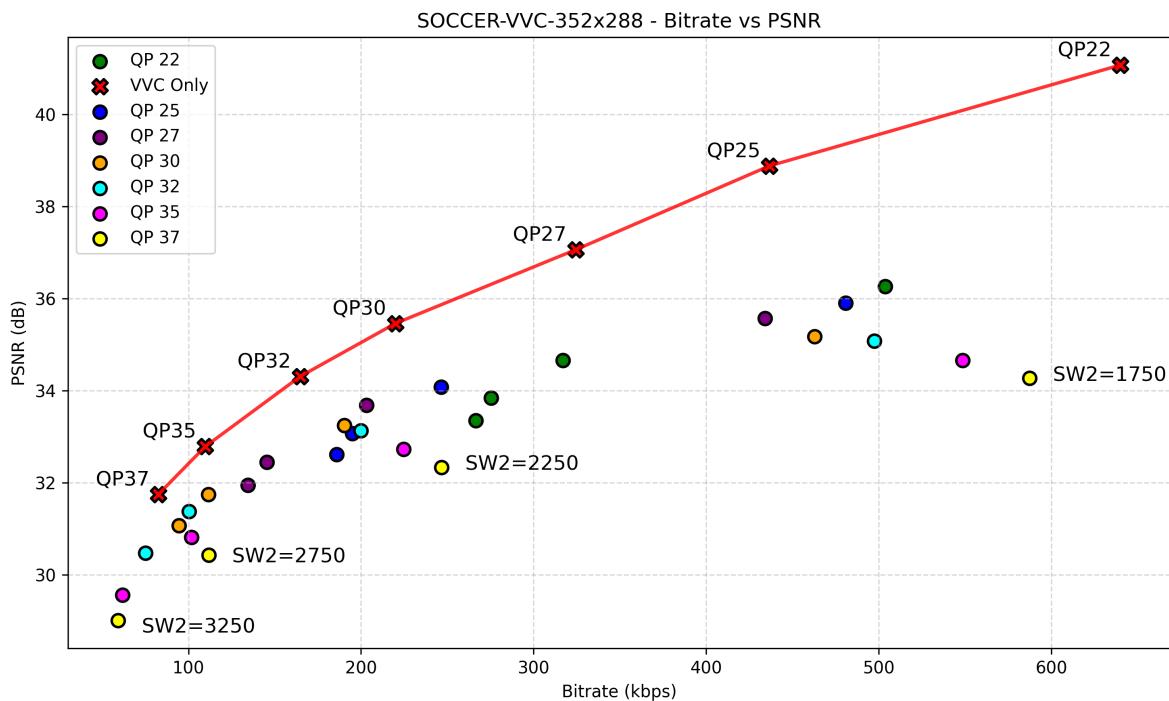


Figura 3.11: Resultados para "SOCCER" em VVC. [3]

Os resultados demonstram que neste caso houve uma vantagem para o uso de somente o VVC, com desempenho superior em todas as configurações testadas. O LCEVC demonstrou que nesta sequência, para os maiores valores de SW2 escolhidos, ele foi capaz de chegar próximo a relação de tamanho e qualidade que o VVC conseguiu, porém não conseguiu superá-lo e foi se distanciando dos pontos obtidos pelo VVC.

3.2.3 Jockey

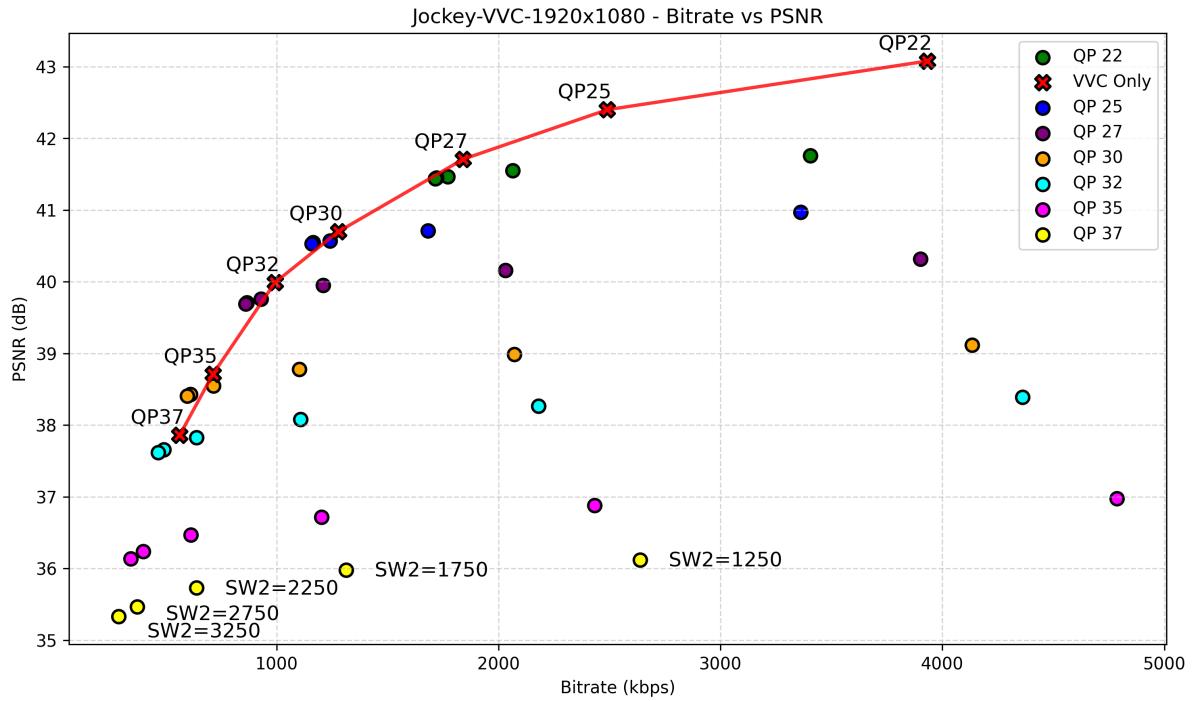


Figura 3.12: Resultados para "Jockey" em VVC. [2]

Para esta sequência, o LCEVC obteve bons resultados acima da curva de eficiência obtida pelo VVC puro. Valores para QPs iguais a 30, 27, 25 se saíram bem e estão acima da curva. QP igual a 22 ficou bastante próximo, mostrando ser um resultado com potencial, onde a qualidade e o tamanho destes arquivos estão próximos ao QP 27 do VVC.

QP	Tamanho (b)	PSNR (dB)	Taxa (kbps)
37	704469	37.86	563.58
35	891197	38.71	712.96
32	1241194	39.99	992.96
30	1598336	40.70	1278.67
27	2300814	41.71	1840.65
25	3111449	42.40	2489.16
22	4916006	43.08	3932.80

Tabela 3.15: Resultados para Jockey em VVC

SW2	QP	Tamanho (b)	Proporção	PSNR (dB)	Taxa (kbps)	Superou
3250	30	747523	2.48%	38.41	598.02	VVC QP37
2750	30	765746	4.80%	38.43	612.60	VVC QP37
3250	27	1075071	1.59%	39.69	860.06	VVC QP35-32
2750	27	1083226	2.33%	39.71	866.58	VVC QP35-32
2250	27	1162481	8.99%	39.76	929.98	VVC QP35-32
3250	25	1449506	2.17%	40.53	1159.60	VVC QP32-30
2750	25	1456646	2.65%	40.55	1165.32	VVC QP32-30

Tabela 3.16: Resultados favoráveis do LCEVC para Jockey em VVC.

3.2.4 City

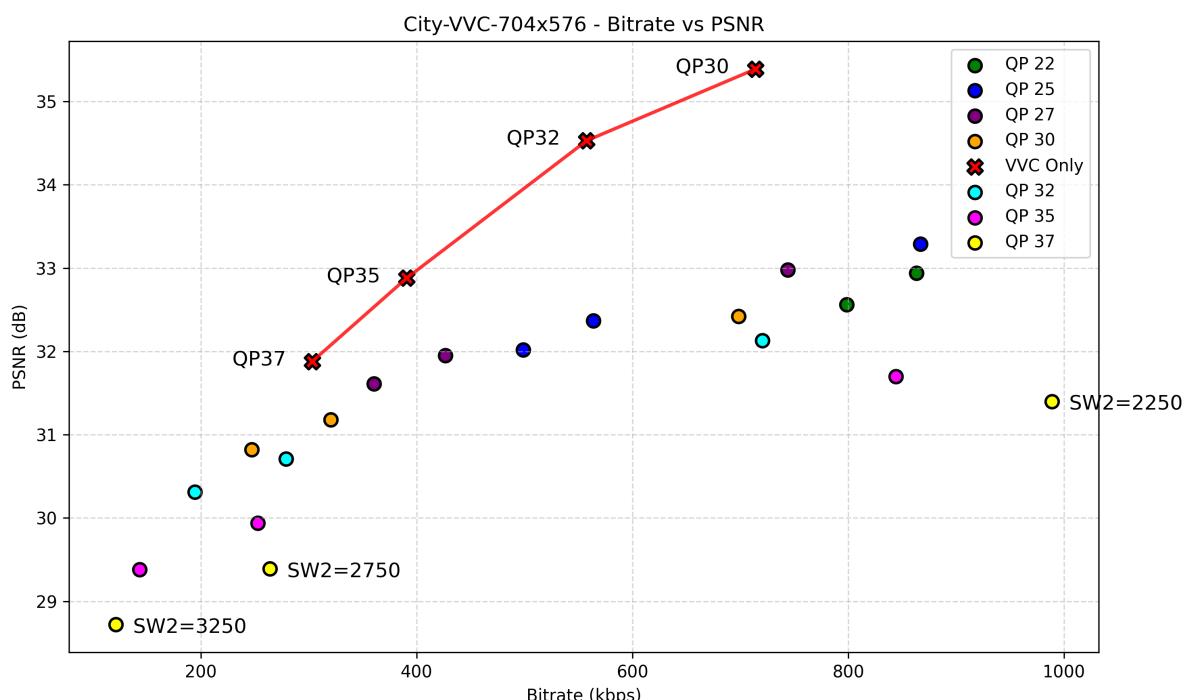


Figura 3.13: Resultados para "City" em VVC. [3]

Em City, o LCEVC não conseguiu superar os resultados obtidos com o VVC. Os resultados aparentam estar seguindo a mesma tendência que os resultados de referência no início, mas começam a decair ao chegarem perto do VVC QP37. Com esse decaimento, se distanciam cada vez mais e não conseguem um resultado satisfatório.

3.2.5 vc-phillips-01

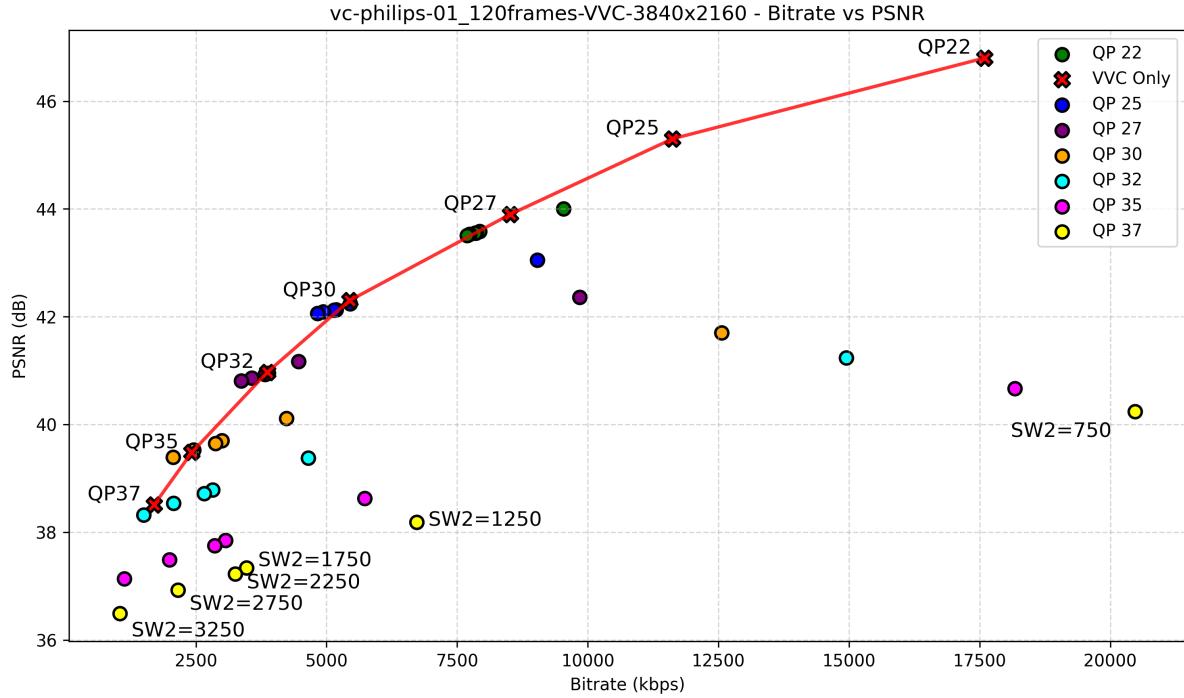


Figura 3.14: Resultados para "vc-phillips-01" em VVC.

Esta sequência obteve uma quantidade considerável de resultados positivos para o LCEVC, para valores de QP iguais 30, 27, 25 e 22. Para o QP 22, observa-se que o LCEVC obteve vários valores em cima da curva de eficiência das sequências de referência, onde pela tendência observada, é possível que os resultados para um QP menor que 22 iria resultar em resultados abaixo da curva de eficiência.

QP	Tamanho (b)	PSNR (dB)	Taxa (kbps)
37	425336	38.51	1701.34
35	604903	39.48	2419.61
32	966774	40.97	3867.10
30	1360536	42.30	5442.14
27	2129169	43.90	8516.68
25	2905023	45.30	11620.09
22	4397908	46.80	17591.63

Tabela 3.17: Resultados para vc-phillips-01 em VVC.

SW2	QP	Tamanho (b)	Proporção	PSNR (dB)	Taxa (kbps)	Superou
3250	30	516071	5.39%	39.39	2064.28	VVC QP37-35
2750	30	613763	20.45%	39.53	2455.05	VVC QP35
3250	27	842020	1.12%	40.81	3368.08	VVC QP35-32
2750	27	891745	6.63%	40.86	3566.98	VVC QP35-32
2250	27	955257	12.84%	40.93	3821.03	VVC QP35-32
1750	27	971277	14.28%	40.95	3885.11	VVC QP35-32
3250	25	1205943	0.50%	42.06	4823.77	VVC QP32-30
2750	25	1234327	2.78%	42.09	4937.31	VVC QP32-30
2250	25	1286043	6.69%	42.12	5144.17	VVC QP32-30
1750	25	1295331	7.36%	42.13	5181.32	VVC QP32-30
3250	22	1922342	0.26%	43.51	7689.37	VVC QP30-27
2750	22	1933946	0.86%	43.53	7735.78	VVC QP30-27
2250	22	1961129	2.24%	43.55	7844.52	VVC QP30-27
1250	22	1982576	3.29%	43.58	7930.30	VVC QP30-27
1750	22	1965174	2.44%	43.55	7860.70	VVC QP30-27

Tabela 3.18: Resultados favoráveis do LCEVC para vc-phillips-01 em VVC.

3.2.6 vc-lcevc-01

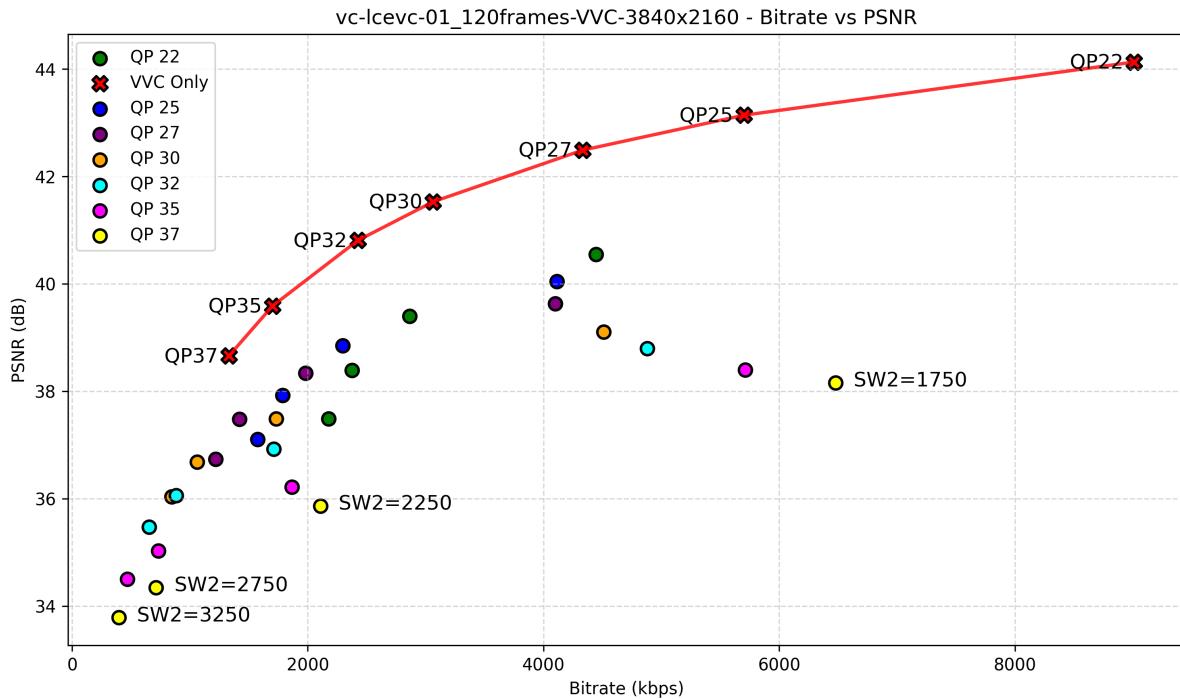


Figura 3.15: Resultados para "vc-lcevc-01" em VVC.

Para esta sequência, o LCEVC não obteve resultados acima da curva de eficiência. Os resultados para o LCEVC ficaram relativamente distantes da curva de referência do VVC. Ela apresenta um mesmo formato de curva, sugerindo que está mantendo a qualidade da mesma maneira que o VVC puro.

3.2.7 vc-globo-05

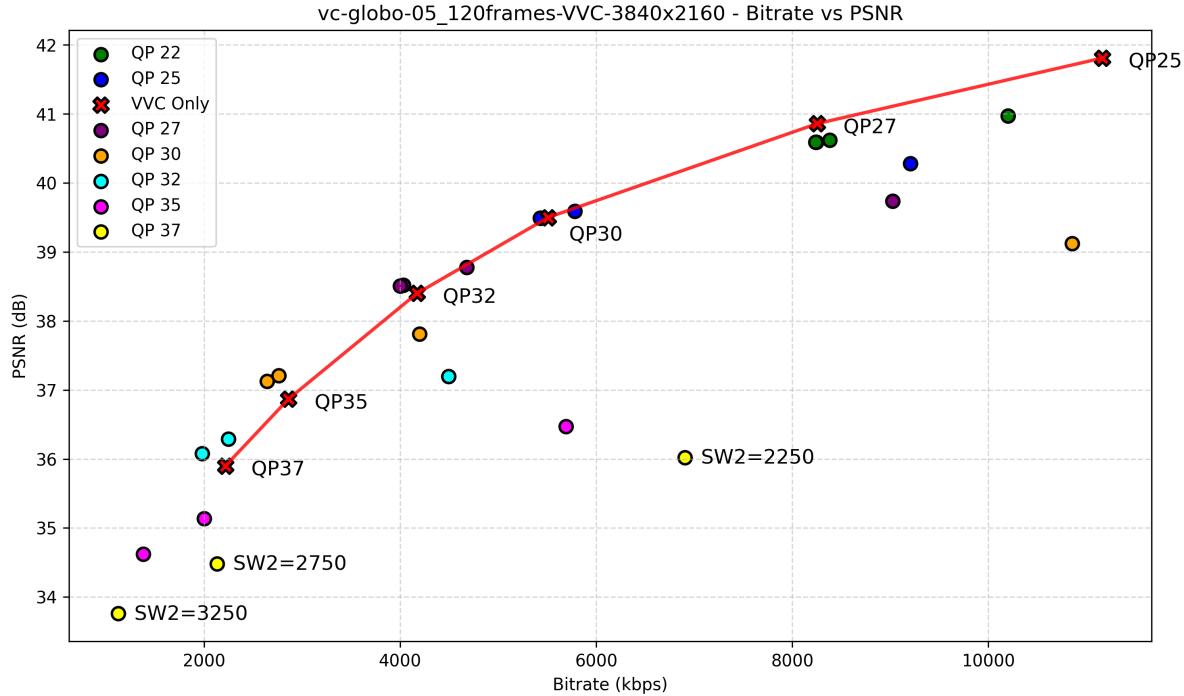


Figura 3.16: Resultados para "vc-globo-05" em VVC [4].

Em *vc-globo-05*, o LCEVC obteve resultados positivos melhores que os resultados obtidos pelo VVC. Os valores de QP que conseguiram resultados acima da curva foram 32, 30, 27 e 25. Para QP igual a 22, os resultados ficaram com o valor de PSNR levemente inferior ao QP 27 do VVC, com a mesma taxa de bits.

QP	Tamanho (b)	PSNR (dB)	Taxa (kbps)
37	554404	35.90	2217.62
35	715402	36.87	2861.61
32	1042958	38.40	4171.83
30	1377368	39.50	5509.47
27	2063711	40.86	8254.84
25	2790684	41.81	11162.74

Tabela 3.19: Resultados para vc-globo-05 em VVC.

SW2	QP	Tamanho (b)	Proporção	PSNR (dB)	Taxa (kbps)	Superou
3250	32	495040	1.35%	36.08	1980.16	VVC QP37
2750	32	561763	13.06%	36.29	2247.05	VVC QP37
3250	30	660706	0.58%	37.13	2642.82	VVC QP35
2750	30	689863	4.78%	37.21	2759.45	VVC QP35
3250	27	1000292	0.31%	38.51	4001.17	VVC QP32
2750	27	1007875	1.06%	38.52	4031.50	VVC QP32
3250	25	1356490	0.44%	39.49	5425.96	VVC QP30
2750	25	1359403	0.66%	39.49	5437.61	VVC QP30

Tabela 3.20: Resultados favoráveis do LCEVC para vc-globo-05 em VVC.

3.3 Visão geral dos resultados

Com os estes resultados obtidos, as tabelas abaixo demonstram a quantidade de pontos favoráveis para o LCEVC para cada sequência.

Sequência	Pontos favoráveis
Bosphorus (1920x1080, AVC)	7
ReadySteadyGo (1920x1080, AVC)	0
Jockey (1920x1080, AVC)	13
SOCCKER (352x288, AVC)	0
City (704x576, AVC)	0
vc-globo-05 (3840x2160, AVC)	5
vc-lcevc-01 (3840x2160, AVC)	5
vc-phillips-01 (3840x2160, AVC)	4
vc-phillips-03 (3840x2160, AVC)	5

Tabela 3.21: Quantidade de pontos acima da curva de eficiência para o AVC.

Sequência	Pontos favoráveis
Bosphorus (1920x1080, VVC)	11
SOCCKER (352x288, VVC)	0
Jockey (1920x1080, VVC)	7
City (704x576, VVC)	0
vc-phillips-01 (3840x2160, VVC)	15
vc-lcevc-01 (3840x2160, VVC)	0
vc-globo-05 (3840x2160, VVC)	8

Tabela 3.22: Quantidade de pontos acima da curva de eficiência para o VVC.

Analizando os resultados obtidos, pode-se observar os seguintes comportamentos:

- O LCEVC apresentou resultados superiores ao codificador base (AVC ou VVC) em diversas sequências, especialmente nas de maior resolução.

- Para sequências de baixa resolução, como SOCCER e City, o LCEVC não demonstrou vantagem significativa em relação ao codificador base.
- Os melhores resultados do LCEVC foram obtidos para valores de QP mais baixos (maior qualidade), onde superou a curva de eficiência dos codificadores de referência.
- Em algumas sequências, o LCEVC manteve a qualidade próxima ao codificador base, mesmo sem superá-lo, indicando potencial para aplicações onde a compatibilidade é importante.
- O número de pontos favoráveis ao LCEVC foi maior nas sequências codificadas com VVC em resoluções UHD (3840x2160).
- O desempenho do LCEVC depende fortemente dos parâmetros escolhidos, especialmente do valor de SW2 e do QP.
- Em geral, o LCEVC se mostrou mais eficiente em cenários de alta qualidade e alta resolução, enquanto em resoluções menores o benefício foi limitado.

Agora, para o parâmetro SW2 do LCEVC, observou-se que para o intervalo escolhido, os maiores valores foram os que mais apareceram dentre os pontos acima da curva de eficiência. A quantidade que cada valor de SW2 apareceram dentre estes pontos foram:

SW2	Ocorrências
3250	36
2750	28
2250	12
1750	3
1250	1

Tabela 3.23: Quantidade de ocorrências de cada valor de SW2 nos resultados favoráveis ao LCEVC

Para poder destacar os parâmetros que conseguiram os resultados positivos, a tabela a seguir demonstra quantas vezes cada SW2 e QP juntos apareceram como pontos acima da curva de eficiência.

SW2	QP	Ocorrências
3250	32	1
3250	30	10
3250	27	10
3250	25	10
3250	22	5
2750	32	1
2750	30	5
2750	27	9
2750	25	8
2750	22	5
2250	32	1
2250	30	2
2250	27	4
2250	25	3
2250	22	2
1750	27	1
1750	25	1
1750	22	1
1250	22	1

Tabela 3.24: Frequência de ocorrência dos pares SW2 e QP para o LCEVC nos pontos favoráveis.

Capítulo 4

Conclusão

Este trabalho apresentou uma análise de qualidade do LCEVC, comparando seus resultados com vídeos que só utilizaram o codificador base utilizado. O LCEVC demonstrou um resultado promissor, onde vários pontos foram superiores aos resultados obtidos pelo codificador da camada base. Com a análise dos resultados obtidos, foi possível perceber que o LCEVC realmente proporciona benefícios de qualidade e tamanho, e com uma complexidade reduzida.

Os resultados mostram que o LCEVC se sai bem com vídeos de resoluções como Full HD (1080p) e 4k, mas não se saiu bem com vídeos com resoluções menores que isso. Dentro dos resultados onde o LCEVC superou os vídeos somente com o codificador base, alguns valores de configuração para o LCEVC apareceram mais, demonstrando serem valores ideias para serem usados. Estes valores foram: **SW2 = 3250 com QP = 30**, **SW2 = 3250 com QP = 27** e **SW2 = 3250 com QP = 25** onde apareceram 10 vezes cada em pontos acima da curva de eficiência. Outros valores que também demonstraram eficiência foram **SW2 = 2750 com QP = 27** e **SW2 = 2750 com QP = 25**. Estes valores apareceram 9 e 8 vezes respectivamente. Esses pares de valores demonstraram ser os melhores candidatos a uma codificação que resulte em um resultado melhor que uma codificação utilizando somente o codificador da camada base.

Com isso, o Low Complexity Enhancement Video Coding mostrou ser capaz de ser mais eficiente que vídeos utilizando somente a camada base, e que é uma opção válida a ser usada junto dos codificadores atuais. Com estes resultados positivos, o LCEVC se mostra uma ótima tecnologia a ser incorporada em aplicações como a TV 3.0, onde o projeto original já demonstra sua viabilidade [6, 7].

4.1 Trabalhos Futuros

Embora este trabalho tenha explorado o LCEVC em combinação com codecs tradicionais, há várias direções futuras que podem ser investigadas:

- Explorar o uso do LCEVC com outros codecs, como HEVC e EVC, para avaliar seu desempenho em diferentes cenários de compressão.
- Investigar a aplicação do LCEVC em contextos de transmissão ao vivo, onde a latência e a eficiência são críticas.
- Analisar o impacto do LCEVC em dispositivos com recursos limitados, como smartphones e dispositivos embarcados, para entender melhor sua viabilidade em cenários de baixa potência.
- Realizar uma análise mais aprofundada da complexidade computacional do LCEVC, comparando-a com outros métodos de aprimoramento de vídeo.
- Investigar a integração do LCEVC com técnicas de aprendizado de máquina para otimização de parâmetros e melhoria da qualidade do vídeo.
- Realizar um estudo mais profundo sobre os parâmetros de codificação do LCEVC, como o SW1 e outras centenas de parâmetros disponíveis, para entender melhor como eles afetam a qualidade e a eficiência da codificação.
- Explorar a possibilidade de utilizar o LCEVC em conjunto com técnicas de super-resolução para melhorar ainda mais a qualidade do vídeo em resoluções mais baixas. Como por exemplo, o uso do Deep Learning Super Sampling (DLSS) ao invés do algoritmo de *Upsampling* do LCEVC.

Referências

- [1] Wikipedia: *Pareto efficient frontier*. https://en.wikipedia.org/wiki/Pareto_front, acesso em 2025-07-27. ix, 10
- [2] Mercat, A., M. Viitanen e J. Vanne: *Uvg dataset: 50/120fps 4k sequences for video codec analysis and development*. Proc. ACM Multimedia Syst. Conf., June 2020. ix, 10, 13, 14, 15, 24, 26
- [3] Xiph.Org: *Xiph.org video test media*. <https://media.xiph.org/video/derf/>, acesso em 2025-05-05. ix, 10, 16, 17, 25, 27
- [4] TV Globo: *Trecho de vídeo da tv globo em formato raw/mov*, 2024. Conteúdo obtido para testes de compressão. Sem distribuição pública. ix, 18, 31
- [5] Syahbana, Yoanda Alim, Herman, Azizah Abdul Rahman e Kamalrulnizam Abu Bakar: *Aligned-psnr (apsnr) for objective video quality measurement (vqm) in video stream over wireless and mobile network*. Em *2011 World Congress on Information and Communication Technologies (WICT)*, Johor, Malaysia, 2011. IEEE. <https://core.ac.uk/download/pdf/228026495.pdf>. x, 5
- [6] SBTVD: *Resultados da avaliação subjetiva da qualidade da codificação de vídeo em tempo real para tv 2.5*. <https://forumsbtvd.org.br/tv-2-5/>, acesso em 2023-07-30. 1, 35
- [7] SBTVD: *Background - terrestrial tv evolution in brazil*. https://forumsbtvd.org.br/tv3_0/, acesso em 2023-07-30. 1, 35
- [8] Carlos Eduardo Cosme Ribeiro: *Gt tv 3.0: Funcionalidades do mpeg-5, low complexity enhancement video coding (lcevc)*. <https://set.org.br/news-revista-da-set/revista/gt-tv-3-0-funcionalidades-do-mpeg-5-low-complexity-enhancement-video-coding-lcevc/>. 1
- [9] V-Nova: *Brazilian tv giant achieves 10mbps uhd with lcevc enhancement*. <https://v-nova.com/press-releases/brazilian-tv-giant-achieves-10mbps-uhd-with-lcevc-enhancement/>, acesso em 2024-09-05. 1
- [10] Miller, Jeremy: *What is raw footage and when to use it*. <https://bxfilms.tv/blog/what-is-raw-footage>, acesso em 2022-09-09. 2
- [11] Morris, O.J.: *Mpeg-2: where did it come from and what is it?* Em *IEE Colloquium on MPEG-2 - What it is and What it isn't*, páginas 1/1–1/5, 1995. 2

- [12] Carlos Eduardo Cosme Ribeiro e Globo: *Panorama dos codificadores de vídeos*. <https://globotech.globo.com/blog/noticia/panorama-dos-codificadores-de-videos.ghtml>, acesso em 2024-08-13. 2
- [13] ITU-T Study Group 16 (VCEG) and ISO/IEC JTC1/SC29/WG11 (MPEG): *ITU-T Recommendation H.264 / ISO/IEC 14496-10: Advanced video coding for generic audiovisual services*. Recommendation / international standard H.264 / ISO/IEC 14496-10, International Telecommunication Union, International Organization for Standardization International Electrotechnical Commission, maio 2003. <https://www.videosurveillance.co.in/H.264.pdf>, Corrigendum 1 added May 7, 2004. 3
- [14] Hamidouche, Wassim, Thibaud Biatek, Mohsen Abdoli, Edouard François, Fernando Pescador, Miloš Radosavljević, Daniel Menard e Mickael Raulet: *Versatile video coding standard: A review from coding tools to consumers deployment*. arXiv preprint, junho 2021. <https://arxiv.org/pdf/2106.14245.pdf>, Revised version (v2) submitted on 6 Nov 2021. 4
- [15] Gupta, Prakash C.: *Data Communications and Computer Networks*. PHI Learning Pvt. Ltd., New Delhi, India, 2006, ISBN 9788120328464. https://books.google.com.br/books?id=-kNn_p6WA38C&pg=PA7. 4
- [16] ISO/IEC JTC 1/SC 29/WG 11 MPEG-5 Part 2 Working Group: *White paper on low complexity enhancement video coding (lcevc)*. White paper MPEG-137, LCEVC (Low Complexity Enhancement Video Coding), janeiro 2022. 5, 6, 7
- [17] Battista, Stefano, Guido Meardi, Simone Ferrara, Lorenzo Ciccarelli, Florian Maurer, Massimo Conti e Simone Orcioni: *Overview of the low complexity enhancement video coding (lcevc) standard*. IEEE Transactions on Circuits and Systems for Video Technology, 32(11):7983–7995, 2022. 6, 7
- [18] Lucas Pena: *Lcevc-enhancement*. https://github.com/lucpena/LCEVC-Enhancement/tree/main/Low_Complexity_Enhancement_Video_Coding, acesso em 2025-07-27. 9, 39
- [19] Emmerich, Michael e André Deutz: *Multicriteria optimization and decision making: Principles, algorithms and case studies*, 2025. <https://arxiv.org/abs/2407.00359>, Página 26. 9
- [20] *LTM Git (requer uma conta MPEG)*. <https://git.mpeg.expert/MPEG/Video/LCEVC/LTM>. 11

Anexo I

Scripts

Todos os códigos criados para este trabalho estão disponíveis no GitHub [18]. Recomenda-se o GitHub para a análise do código, devido a organização dos arquivos. Para garantir a preservação destes códigos, eles serão anexados abaixo.

Os scripts foram divididos em duas baterias de codificação diferentes. Uma para AVC e uma para VVC. Cada um possui seu *main* que irá chamar todos os códigos necessários de maneira paralela, para agilizar a geração dos resultados. Entretanto, este script foi executado em um processador com vários núcleos que se beneficia com a paralelização. Caso for executar este script em um processador com menos recursos, recomenda-se a alteração do script para que ele seja executado sequencialmente, ou diminuir a quantidade de processos criados.

O script cria uma pasta chamada *avc_only*, onde é necessário colar nesta pasta o *encoder.cfg* do codificar AVC. Também é possível alterar o script e passar o endereço do *encoder.cfg*.

Coloque todos estes arquivos na mesma pasta.

I.1 AVC

main-avc.sh

```
#!/bin/bash

echo -en "\nInicializando Codificacao em AVC... "
for i in {5..1}; do
    sleep 1
    echo -n "${i} "
done
echo -e "\n"
```

```

# === Configuracoes compartilhadas ===
# InputFile="/mnt/md0/lucpena/videos/Bosphorus_1920x1080_120fps_420_8bit.yuv"
# InputFile="/mnt/md0/lucpena/videos/SOCCER_352x288_30_orig_02.yuv"
# InputFile="/mnt/md0/lucpena/videos/RaceNight_3840x2160_50fps_8bit.yuv"
# InputFile="/mnt/md0/lucpena/videos/ReadySteadyGo_1920x1080_120fps_420_8bit.
yuv"
# InputFile="/mnt/md0/lucpena/videos/Jockey_1920x1080_120fps_420_8bit.yuv"
# InputFile="/mnt/md0/lucpena/videos/city_704x576_yuv420p_60fps_600frames.yuv"
InputFile="/mnt/md0/lucpena/videos/SBTVD/YUV/vc-globo-05_120frames_420p.yuv"

# VIDEO_NAME="Bosphorus"
# VIDEO_NAME="SOCCER"
# VIDEO_NAME="RaceNight"
# VIDEO_NAME="ReadySteadyGo"
# VIDEO_NAME="Jockey"
# VIDEO_NAME="City"
VIDEO_NAME="vc-globo-05_120frames"

# WIDTH=1920
# HEIGHT=1080
# FPS=60
# NUM_FRAMES=600

## Soocer
# WIDTH=352
# HEIGHT=288
# FPS=30
# NUM_FRAMES=300

## City
# WIDTH=704
# HEIGHT=576
# FPS=60
# NUM_FRAMES=600

## SBTVD
WIDTH=3840
HEIGHT=2160
FPS=60

```

```

NUM_FRAME=120

SW2=250
SW2_max=3250
FORMAT="yuv420p"

BaseEncoder="avc"
Step=500

MAIN_TIME=$(date +"%Y-%m-%d_%H-%M-%S")

QP_list=(22 25 27 30 32 35 37)

ENCODER_SCRIPT="avc_template.sh"
SCRIPT_DIR=$(dirname "$(readlink -f "$0")")

BaseEncoderUpper=$(echo "$BaseEncoder" | tr '[:lower:]' '[:upper:]')
SizesFile="$SCRIPT_DIR/${VIDEO_NAME}-${BaseEncoderUpper}-${WIDTH}x${HEIGHT}-${
MAIN_TIME}.csv"

# Loop para LCEVC
for QP in "${QP_list[@]}"; do
    DIR="QP${QP}"
    mkdir -p "$DIR"
    cp "$ENCODER_SCRIPT" "$DIR/run.sh"

    # Cria script com todas as variaveis embutidas
    cat > "$DIR/config.sh" <<EOF
#!/bin/bash

MAIN_TIME="$MAIN_TIME"
InputFile="$InputFile"
VIDEO_NAME="$VIDEO_NAME"
WIDTH=$WIDTH
HEIGHT=$HEIGHT
FORMAT="$FORMAT"
FPS=$FPS
NUM_FRAME=$NUM_FRAME
SW2=$SW2
SW2_max=$SW2_max
Step=$Step

```

```

BaseEncoder="$BaseEncoder"
QP=$QP
BaseEncoderUpper="$BaseEncoderUpper"
SizesFile="$SizesFile"
EOF

# Cria o SizesFile se nao existir
if [ ! -f "$SizesFile" ]; then
    echo "file,sw2,qp,lv_size,base_size,enhance_size,enhancement_ratio,
psnr_avg,bitrate_kbps" > "$SizesFile"
fi

chmod +x "$DIR/run.sh"
chmod +x "$DIR/config.sh"

# Executa em segundo plano
(cd "$DIR" && ./run.sh) &
sleep 0.5
done

# === Script so com AVC ===
AVC_ONLY_DIR="$SCRIPT_DIR/avc_only"
AVC_TEMPLATE="$SCRIPT_DIR/avc_only_template.sh"
mkdir -p "$AVC_ONLY_DIR"

for QP in "${QP_list[@]}"; do
    QPDIR="$AVC_ONLY_DIR/QP${QP}"
    mkdir -p "$QPDIR"

    # Copia template e torna executavel
    cp "$AVC_TEMPLATE" "$QPDIR/run.sh"
    chmod +x "$QPDIR/run.sh"

    # Config file
    cat > "$QPDIR/config.sh" <<EOF
#!/bin/bash
MAIN_TIME="$MAIN_TIME"
InputFile="$InputFile"
VIDEO_NAME="$VIDEO_NAME"
WIDTH=$WIDTH

```

```

HEIGHT=$HEIGHT
FORMAT="$FORMAT"
FPS=$FPS
NUM_FRAMES=$NUM_FRAMES
BaseEncoder="$BaseEncoder"
QP=$QP
SizesFile="$SizesFile"
EOF
chmod +x "$QPDIR/config.sh"

# Dispara em segundo plano
(cd "$QPDIR" && ./run.sh "$QP") &
sleep 0.5
done

# === Script com Down/Up sampling ===
DIR="resample"
mkdir -p "$DIR"
cp "resample_template.sh" "$DIR/run.sh"

cat > "$DIR/config.sh" <<EOF
#!/bin/bash
MAIN_TIME="$MAIN_TIME"
InputFile="$InputFile"
VIDEO_NAME="$VIDEO_NAME"
WIDTH=$WIDTH
HEIGHT=$HEIGHT
FORMAT="$FORMAT"
FPS=$FPS
NUM_FRAMES=$NUM_FRAMES
SizesFile="$SizesFile"
EOF
chmod +x "$DIR/*.sh

(cd "$DIR" && ./run.sh) &

wait
echo -e "\n-----\n Finish \n-----\n"

```

avc_only_template.sh

```
#!/bin/bash

source ./config.sh

QP="$1"
if [[ -z "$QP" ]]; then
    echo "Erro: QP nao fornecido!"
    exit 1
fi

BaseEncoderUpper=$(echo "$BaseEncoder" | tr '[lower:]' '[upper:]')
SCRIPT_PATH=$(readlink -f "$0")
SCRIPT_DIR=$(dirname "$SCRIPT_PATH")
STATUS_FILE="$SCRIPT_DIR/status-${QP}.log"
ENC_DIR="/mnt/md0/lucpena/output/main/output"

rm -f "$STATUS_FILE"
echo -e "Iniciando o script AVC Only\n\n" >> "$STATUS_FILE"

SizesFileAVC="$SCRIPT_DIR/${VIDEO_NAME}-${BaseEncoderUpper}-only.csv"

# Checa se o CSV ja existe, se nao, cria com o cabecalho
if [ ! -f "$SizesFileAVC" ]; then
    echo "file,sw2,qp,lvc_size,base_size,enhance_size,enhancement_ratio,
psnr_avg,bitrate_kbps" > "$SizesFileAVC"
fi

JMEncoder="/home/lucpena/apps/LTM/_build_linux/external_codecs/JM/lencod"
FFMPEG="/home/lucpena/apps/FFMPEG-VMAF/ffmpeg"

TIMESTAMP=$(date +"%Y-%m-%d_%H-%M-%S")
OutputFile="$ENC_DIR/${VIDEO_NAME}_${BaseEncoderUpper}-Only_QP${QP}.264"
OutputFileName="${OutputFile##*/}"
OutputYUVFile="${OutputFile%.264}.yuv"
LogFile="$SCRIPT_DIR/${VIDEO_NAME}_${BaseEncoderUpper}-Only_QP${QP}.log"
CONFIG="/mnt/md0/lucpena/output/main/all/avc_only/encoder.cfg"

echo -e "\n==== Codificando somente AVC QP=${QP} ===\n" | tee -a "$STATUS_FILE"
```

```

echo "> Codificando com QP=$QP" >> "$STATUS_FILE"

# Codificando para AVC
${JMEncoder} -d "$CONFIG" -p InputFile="$InputFile" -p OutputFile="$OutputFile"
-p SourceWidth=$WIDTH -p SourceHeight=$HEIGHT -p FrameRate=$FPS -p
FramesToBeEncoded=$NUM_FRAMES -p YUVFormat=1 -p QPISlice=$QP -p QPPSlice=$((QP + 1)) -p QPBSlice=$((QP + 1)) -p IntraPeriod=64 2>&1 | tee "$LogFile"

# Convertendo o arquivo de saida para YUV para o PSNR
$FFMPEG -i "$OutputFile" -pix_fmt "$FORMAT" -f rawvideo "$OutputYUVFile" -y >>
"$STATUS_FILE"

# Calculando PSNR
$FFMPEG -s:v "${WIDTH}x${HEIGHT}" -pix_fmt "$FORMAT" -i "$InputFile" \
-s:v "${WIDTH}x${HEIGHT}" -pix_fmt "$FORMAT" -i "$OutputYUVFile" \
-lavfi psnr="stats_file=psnr.log" -f null - >> "$STATUS_FILE"

psnr_avg=$(tail -n 1 psnr.log | awk -F'psnr_avg:' '{print $2}' | awk '{print $1}')
rm psnr.log >> "$STATUS_FILE"

output_file_size=$(stat -c%s "$OutputFile")
bitrate_kbps=$(awk "BEGIN { printf \"%.\.2f\", ($output_file_size * 8 * $FPS) / ($NUM_FRAMES * 1000) }")

[[ -z "$psnr_avg" ]] && psnr_avg="N/A"

# Salva no CSV principal e no local
echo "${OutputFileName},264,$QP,$output_file_size,0,0,0%,${psnr_avg}, ${bitrate_kbps}" >> "$SizesFileAVC"
echo "${OutputFileName},264,$QP,$output_file_size,0,0,0%,${psnr_avg}, ${bitrate_kbps}" >> "$SizesFile"

echo "QP=$QP | PSNR=${psnr_avg} dB | Bitrate=${bitrate_kbps} kbps" >> "$STATUS_FILE"

echo -e "\n\n AVC Only finalizado!" | tee -a "$STATUS_FILE"

```

avc_template.sh

```
#!/bin/bash
source ./config.sh

SCRIPT_PATH=$(readlink -f "$0")
SCRIPT_DIR=$(dirname "$SCRIPT_PATH")

ModelEncoder="/home/lucpena/apps/LTM/_build_linux/ModelEncoder"
ModelDecoder="/home/lucpena/apps/LTM/_build_linux/ModelDecoder"
FFMPEG="/home/lucpena/apps/FFMPEG-VMAF/ffmpeg"

MAIN_DIR="/mnt/md0/lucpena/output/main"
BASES_DIR="$MAIN_DIR/bases"
DEC_DIR="$MAIN_DIR/dec"
LOGS_DIR="$MAIN_DIR/logs"
BIN_DIR="$MAIN_DIR/bin"
RESULTS_DIR="$MAIN_DIR/results"
STATUS_FILE="$SCRIPT_DIR/status.log"

LOCAL_CSV="$SCRIPT_DIR/${VIDEO_NAME}-qp${QP}.csv"

# Checa se o CSV ja existe, se nao, cria com o cabecalho
if [ ! -f "$LOCAL_CSV" ]; then
    echo "file,sw2,qp,lvc_size,base_size,enhance_size,enhancement_ratio,
psnr_avg,bitrate_kbps" > "$LOCAL_CSV"
fi

rm -f "$STATUS_FILE"
echo -e "\n==== Iniciando testes para QP=${QP} ===\n" | tee -a "$STATUS_FILE"

rm -f $SCRIPT_DIR/*.yuv $SCRIPT_DIR/psnr.log 2>> "$STATUS_FILE"

BASE_NAME="${VIDEO_NAME}-qp${QP}-qp${BaseEnconderUpper}.bin"
if [[ ! -f "${BIN_DIR}/${BASE_NAME}" ]]; then
    echo -e "\nO arquivo ${BASE_NAME} nao existe, criando um novo..." | tee -a
    "$STATUS_FILE"
    ${ModelEncoder} \
        --width=${WIDTH} --height=${HEIGHT} --format=${FORMAT} --fps=${FPS}
        --qp=${QP} \
```

```

--input_file="${InputFile}" \
--output_file="$DEC_DIR/base.lvc" --base_encoder=${BaseEncoder}
--encapsulation=nal \
--cq_step_width_loq_0=${SW2} --keep_base \
2>&1 | tee -a "$LOGS_DIR/${VIDEO_NAME}_base-gen_QP${QP}_${(date +"%Y-%m-%d_%H-%M-%S")}.log"

mv "$SCRIPT_DIR/base.bin" "${BIN_DIR}/${BASE_NAME}"

echo -e "Base ${BASE_NAME} criada com sucesso." | tee -a "$STATUS_FILE"
else
    echo -e "Base ${BASE_NAME} ja existente, reutilizando." | tee -a "$STATUS_FILE"
fi

for (( current_sw2=SW2; current_sw2<=SW2_max; current_sw2+=Step )); do
    TIMESTAMP=$(date +"%Y-%m-%d_%H-%M-%S")
    OutputFile="$DEC_DIR/${VIDEO_NAME}_LCEVC_${BaseEncoderUpper}_SW2-${current_sw2}_QP${QP}_${TIMESTAMP}.lvc"
   LogFile="$LOGS_DIR/${VIDEO_NAME}_LCEVC_${BaseEncoderUpper}_SW2-${current_sw2}_QP${QP}_${TIMESTAMP}.log"

    echo -e "\n Rodando encode com SW2 = $current_sw2, QP = $QP" | tee -a "$STATUS_FILE"

    ${ModelEncoder} \
        --width=${WIDTH} --height=${HEIGHT} --format=${FORMAT} --fps=${FPS}
    --qp=${QP} \
        --input_file="${InputFile}" \
        --output_file="${OutputFile}" \
        --base="${BIN_DIR}/${BASE_NAME}" \
        --base_encoder=${BaseEncoder} --encapsulation=nal \
        --cq_step_width_loq_0=${current_sw2} \
        2>&1 | tee -a "${LogFile}"

    DecodedFile="$DEC_DIR/${VIDEO_NAME}_Dec_${BaseEncoderUpper}_LCEVC_SW2-${current_sw2}_QP${QP}_${TIMESTAMP}.yuv"
    DecodedLog="$LOGS_DIR/${VIDEO_NAME}_Dec_LCEVC_${BaseEncoderUpper}_SW2-${current_sw2}_QP${QP}_${TIMESTAMP}.log"

```

```

${ModelDecoder} \
    --base_encoder=${BaseEncoder} --fps=${FPS} \
    --input_file="${OutputFile}" \
    --output_file="${DecodedFile}" \
    --keep_base=true \
    2>&1 | tee -a "${DecodedLog}"

base_file=$(find $SCRIPT_DIR -maxdepth 1 -type f -name "*base.es" | head -n 1)
enhance_file=$(find $SCRIPT_DIR -maxdepth 1 -type f -name "*enhancement.es" | head -n 1)

if [[ -f "$base_file" ]]; then
    base_size=$(stat -c%s "$base_file")
    # mv "$base_file" "$BASES_DIR/${VIDEO_NAME}_Base_LCEVC_${BaseEncoderUpper}_SW2-${current_sw2}_QP${QP}_${TIMESTAMP}.es"
    rm "$base_file"
else
    base_size=0
    echo "base.es nao encontrado!" | tee -a "$STATUS_FILE"
fi

if [[ -f "$enhance_file" ]]; then
    enhance_size=$(stat -c%s "$enhance_file")
    # mv "$enhance_file" "$BASES_DIR/${VIDEO_NAME}_Enha_LCEVC_${BaseEncoderUpper}_SW2-${current_sw2}_QP${QP}_${TIMESTAMP}.es"
    rm "$enhance_file"
else
    enhance_size=0
    echo "enhancement.es nao encontrado!" | tee -a "$STATUS_FILE"
fi

output_file_size=$(stat -c%s "$OutputFile")
if (( output_file_size > 0 )); then
    enhancement_ratio=$(awk "BEGIN { printf \"%.4f\", $enhance_size / $output_file_size * 100 }")
    bitrate_kbps=$(awk "BEGIN { printf \"%.2f\", ($output_file_size * 8 * $FPS) / ($NUM_FRAMES * 1000) }")
else
    enhancement_ratio=0

```

```

bitrate_kbps=0
fi

echo -e "\n\t Calculando PSNR..." | tee -a "$STATUS_FILE"
$FFMPEG \
-s:v "${WIDTH}x${HEIGHT}" -pix_fmt "${FORMAT}" -i "${InputFile}" \
-s:v "${WIDTH}x${HEIGHT}" -pix_fmt "${FORMAT}" -i "${DecodedFile}" \
-lavfi psnr="stats_file=${SCRIPT_DIR}/psnr.log" -f null - 2> /dev/null

psnr_avg=$(tail -n 1 ${SCRIPT_DIR}/psnr.log | awk -F'psnr_avg:' '{print $2}' | awk '{print $1}')
rm -f ${SCRIPT_DIR}/psnr.log >> "$STATUS_FILE"

if [[ -z "$psnr_avg" ]]; then
    psnr_avg="N/A"
fi

FileName="${OutputFile##*/}"
echo "${FileName},${current_sw2},${QP},${output_file_size},${base_size},${enhance_size},${enhancement_ratio}%,${psnr_avg},${bitrate_kbps}" | tee -a "$STATUS_FILE" >> "$SizesFile"
echo "${FileName},${current_sw2},${QP},${output_file_size},${base_size},${enhance_size},${enhancement_ratio}%,${psnr_avg},${bitrate_kbps}" >> "$LOCAL_CSV"

echo -e "\n\t SW2=${current_sw2} enhancement_ratio=${enhancement_ratio}% | \
psnr_avg=${psnr_avg} dB | bitrate=${bitrate_kbps} kbps\n" | tee -a "$STATUS_FILE"

rm -f ${SCRIPT_DIR}/*temp* 2>> "$STATUS_FILE"
done

echo -e "\n Script para QP=${QP} finalizado com sucesso!" | tee -a "$STATUS_FILE"

```

I.2 VVC

main-vvc.sh

```

#!/bin/bash

echo -en "\nInicializando Codificacao em VVC... "
for i in {5..1}; do
    sleep 1
    echo -n "${i} "
done
echo -e "\n"

# === Configuracoes compartilhadas ===
# InputFile="/mnt/md0/lucpena/videos/Bosphorus_1920x1080_120fps_420_8bit.yuv"
# InputFile="/mnt/md0/lucpena/videos/SOCCER_352x288_30_orig_02.yuv"
# InputFile="/mnt/md0/lucpena/videos/RaceNight_3840x2160_50fps_8bit.yuv"
# InputFile="/mnt/md0/lucpena/videos/ReadySteadyGo_1920x1080_120fps_420_8bit.yuv"
# InputFile="/mnt/md0/lucpena/videos/Jockey_1920x1080_120fps_420_8bit.yuv"
# InputFile="/mnt/md0/lucpena/videos/city_704x576_yuv420p_60fps_600frames.yuv"
InputFile="/mnt/md0/lucpena/videos/SBTVD/YUV/vc-globo-05_120frames_420p.yuv"

# VIDEO_NAME="Bosphorus"
# VIDEO_NAME="SOCCER"
# VIDEO_NAME="RaceNight"
# VIDEO_NAME="ReadySteadyGo"
# VIDEO_NAME="Jockey"
# VIDEO_NAME="City"
VIDEO_NAME="vc-globo-05_120frames"

# WIDTH=1920
# HEIGHT=1080
# FPS=60
# NUM_FRAMES=600

## Soocer
# WIDTH=352
# HEIGHT=288
# FPS=30
# NUM_FRAMES=300

## City
# WIDTH=704

```

```

# HEIGHT=576
# FPS=60
# NUM_FRAMES=600

## vc-globo-05
WIDTH=3840
HEIGHT=2160
FPS=60
NUM_FRAMES=120

SW2=250
SW2_max=3250
FORMAT="yuv420p"

BaseEncoder="vvc"
Step=500

MAIN_TIME=$(date +"%Y-%m-%d_%H-%M-%S")

QP_list=(22 25 27 30 32 35 37)

ENCODER_SCRIPT="vvc_template.sh"
SCRIPT_DIR=$(dirname "$(readlink -f "$0")")

BaseEncoderUpper=$(echo "$BaseEncoder" | tr '[lower:]' '[upper:]')
SizesFile="$SCRIPT_DIR/${VIDEO_NAME}-${BaseEncoderUpper}-${WIDTH}x${HEIGHT}-${
    MAIN_TIME}.csv"

# Loop para LCEVC
for QP in "${QP_list[@]}"; do
    DIR="QP${QP}"
    mkdir -p "$DIR"
    cp "$ENCODER_SCRIPT" "$DIR/run.sh"

    # Cria script com todas as variaveis embutidas
    cat > "$DIR/config.sh" <<EOF
#!/bin/bash
MAIN_TIME="$MAIN_TIME"
InputFile="$InputFile"
VIDEO_NAME="$VIDEO_NAME"

```

```

WIDTH=$WIDTH
HEIGHT=$HEIGHT
FORMAT="$FORMAT"
FPS=$FPS
NUM_FRAMES=$NUM_FRAMES
SW2=$SW2
SW2_max=$SW2_max
Step=$Step
BaseEncoder="$BaseEncoder"
QP=$QP
BaseEncoderUpper="$BaseEncoderUpper"
SizesFile="$SizesFile"
EOF

# Cria o SizesFile se nao existir
if [ ! -f "$SizesFile" ]; then
    echo "file,sw2,qp,lv_size,base_size,enhance_size,enhancement_ratio,
psnr_avg,bitrate_kbps" > "$SizesFile"
fi

chmod +x "$DIR/run.sh"
chmod +x "$DIR/config.sh"

# Executa em segundo plano
(cd "$DIR" && ./run.sh) &
sleep 2
done

# === Script so com VVC ===
VVC_ONLY_DIR="$SCRIPT_DIR/vvc_only"
VVC_TEMPLATE="$SCRIPT_DIR/vvc_only_template.sh"
mkdir -p "$VVC_ONLY_DIR"

for QP in "${QP_list[@]}"; do
    QPDIR="$VVC_ONLY_DIR/QP${QP}"
    mkdir -p "$QPDIR"

    # Copia template e torna executavel
    cp "$VVC_TEMPLATE" "$QPDIR/run.sh"
    chmod +x "$QPDIR/run.sh"

```

```

# Config file
cat > "$QPDIR/config.sh" <<EOF
#!/bin/bash
MAIN_TIME="$MAIN_TIME"
InputFile="$InputFile"
VIDEO_NAME="$VIDEO_NAME"
WIDTH=$WIDTH
HEIGHT=$HEIGHT
FORMAT="$FORMAT"
FPS=$FPS
NUM_FRAMES=$NUM_FRAMES
BaseEncoder="$BaseEncoder"
QP=$QP
SizesFile="$SizesFile"
EOF
chmod +x "$QPDIR/config.sh"

# Dispara em segundo plano
(cd "$QPDIR" && ./run.sh "$QP") &
sleep 2
done

# === Script com Down/Up sampling ===
DIR="resample"
mkdir -p "$DIR"
cp "resample_template.sh" "$DIR/run.sh"

cat > "$DIR/config.sh" <<EOF
#!/bin/bash
MAIN_TIME="$MAIN_TIME"
InputFile="$InputFile"
VIDEO_NAME="$VIDEO_NAME"
WIDTH=$WIDTH
HEIGHT=$HEIGHT
FORMAT="$FORMAT"
FPS=$FPS
NUM_FRAMES=$NUM_FRAMES
SizesFile="$SizesFile"
EOF

```

```

chmod +x "$DIR/*.sh

(cd "$DIR" && ./run.sh) &

wait

echo -e "\n-----\n Finish \n-----\n"

```

vvc_only_template.sh

```

#!/bin/bash

source ./config.sh

QP="$1"
if [[ -z "$QP" ]]; then
    echo "Erro: QP nao fornecido!"
    exit 1
fi

BaseEncoderUpper=$(echo "$BaseEncoder" | tr '[:lower:]' '[:upper:]')
SCRIPT_PATH="$(readlink -f "$0")"
SCRIPT_DIR="$(dirname "$SCRIPT_PATH")"
STATUS_FILE="$SCRIPT_DIR/status-${QP}.log"
ENC_DIR="/mnt/md0/lucpena/output/main/output"

rm -f "$STATUS_FILE"
echo -e "Iniciando o script VVC Only\n\n" >> "$STATUS_FILE"

SizesFileVVC="$SCRIPT_DIR/${VIDEO_NAME}-${BaseEncoderUpper}-only.csv"

# Checa se o CSV ja existe, se nao, cria com o cabecalho
if [ ! -f "$SizesFileVVC" ]; then
    echo "file,sw2,qp,lvc_size,base_size,enhance_size,enhancement_ratio,
psnr_avg,bitrate_kbps" > "$SizesFileVVC"
fi

```

```

VVC_Encoder="/home/lucpena/apps/LTM/_build_linux/external_codecs/VTM/EncoderApp
"
FFMPEG="/home/lucpena/apps/FFMPEG-VMAF/ffmpeg"

TIMESTAMP=$(date +"%Y-%m-%d_%H-%M-%S")
OutputFile="$ENC_DIR/${VIDEO_NAME}_${BaseEncoderUpper}-Only_QP${QP}.bin"
OutputFileName="${OutputFile##*/}"
OutputYUVFile="${OutputFile%.bin}.yuv"
LogFile="$SCRIPT_DIR/${VIDEO_NAME}_${BaseEncoderUpper}-Only_QP${QP}.log"

CONFIG="/home/lucpena/apps/LTM/_build_linux/external_codecs/VTM/
vtm_encoder_randomaccess.cfg"

echo -e "\n==== Codificando somente VVC QP=${QP} ===\n" | tee -a "$STATUS_FILE"
echo "Codificando com QP=$QP" >> "$STATUS_FILE"

# Codificando para VVC
$VVC_Encoder -c "$CONFIG" --InputFile=${InputFile} --BitstreamFile=${OutputFile} --ReconFile=recon.yuv --SourceWidth=${WIDTH} --SourceHeight=${HEIGHT} --InputBitDepth=8 --OutputBitDepth=8 --InternalBitDepth=8 --FrameRate=${FPS} --QP=${QP} --FramesToBeEncoded=${NUM_FRAMES} --ConformanceWindowMode=1 --InputChromaFormat=420


# Decodificando o VVC para YUV
VVC_Decoder="/home/lucpena/apps/LTM/_build_linux/external_codecs/VTM/DecoderApp
"
FinalFile="${OutputFile%.bin}-recon.yuv"

${VVC_Decoder} -b ${OutputFile} -o ${OutputYUVFile} >> "$STATUS_FILE"

# Calculando PSNR
$FFMPEG -s:v "${WIDTH}x${HEIGHT}" -pix_fmt "$FORMAT" -i "$InputFile" \
-s:v "${WIDTH}x${HEIGHT}" -pix_fmt "$FORMAT" -i "$OutputYUVFile" \
-lavfi psnr="stats_file=psnr.log" -f null - >> "$STATUS_FILE"

psnr_avg=$(tail -n 1 psnr.log | awk -F'psnr_avg:' '{print $2}' | awk '{print $1}')
rm psnr.log >> "$STATUS_FILE"

```

```

output_file_size=$(stat -c%s "$OutputFile")
bitrate_kbps=$(awk "BEGIN { printf "%.2f\", ($output_file_size * 8 * $FPS) / (
$NUM_FRAMES * 1000) }")

[[ -z "$psnr_avg" ]] && psnr_avg="N/A"

# Salva no CSV principal e no local
echo "${OutputFileName},265,${QP},${output_file_size},0,0,0%,${psnr_avg},${{
bitrate_kbps}" >> "$SizesFileVVC"
echo "${OutputFileName},265,${QP},${output_file_size},0,0,0%,${psnr_avg},${{
bitrate_kbps}" >> "$SizesFile"

echo " QP=$QP | PSNR=${psnr_avg} dB | Bitrate=${bitrate_kbps} kbps" >> "
$STATUS_FILE"

echo -e "\n\n VVC Only com QP${QP} finalizado!" | tee -a "$STATUS_FILE"

```

vvc_template.sh

```

#!/bin/bash
source ./config.sh

SCRIPT_PATH="$(readlink -f "$0")"
SCRIPT_DIR="$(dirname "$SCRIPT_PATH")"

ModelEncoder="/home/lucpena/apps/LTM/_build_linux/ModelEncoder"
ModelDecoder="/home/lucpena/apps/LTM/_build_linux/ModelDecoder"
FFMPEG="/home/lucpena/apps/FFMPEG-VMAF/ffmpeg"

MAIN_DIR="/mnt/md0/lucpena/output/main"
BASES_DIR="$MAIN_DIR/bases"
DEC_DIR="$MAIN_DIR/dec"
LOGS_DIR="$MAIN_DIR/logs"
BIN_DIR="$MAIN_DIR/bin"
RESULTS_DIR="$MAIN_DIR/results"
STATUS_FILE="$SCRIPT_DIR/status.log"

LOCAL_CSV="$SCRIPT_DIR/${VIDEO_NAME}-${BaseEncoderUpper}-QP${QP}.csv"

```

```

# Checa se o CSV ja existe, se nao, cria com o cabecalho
if [ ! -f "$LOCAL_CSV" ]; then
    echo "file,sw2,qp,lvc_size,base_size,enhance_size,enhancement_ratio,
psnr_avg,bitrate_kbps" > "$LOCAL_CSV"
fi

rm -f "$STATUS_FILE"
echo -e "\n==== Iniciando testes para QP=${QP} ===\n" | tee -a "$STATUS_FILE"

rm -f $SCRIPT_DIR/*.yuv $SCRIPT_DIR/psnr.log 2>> "$STATUS_FILE"

BASE_NAME="${VIDEO_NAME}-QP${QP}-${BaseEncoderUpper}.bin"
if [[ ! -f "${BIN_DIR}/${BASE_NAME}" ]]; then
    echo -e "\n0 arquivo a ${BASE_NAME} nao existe, criando um novo..." | tee -a
"$STATUS_FILE"
${ModelEncoder} \
    --width=${WIDTH} --height=${HEIGHT} --format=${FORMAT} --fps=${FPS}
--qp=${QP} \
    --input_file="${InputFile}" \
    --output_file="$DEC_DIR/base.lvc" --base_encoder=${BaseEncoder}
--encapsulation=nal \
    --cq_step_width_loq_0=${SW2} --keep_base \
    2>&1 | tee -a "$LOGS_DIR/${VIDEO_NAME}_base-gen_QP${QP}_${(date +"%Y-%m-
%d_%H-%M-%S")}.log"

mv "$SCRIPT_DIR/base.bin" "${BIN_DIR}/${BASE_NAME}"

echo -e "Base ${BASE_NAME} criada com sucesso." | tee -a "$STATUS_FILE"
else
    echo -e "Base ${BASE_NAME} ja existente, reutilizando." | tee -a "
$STATUS_FILE"
fi

for (( current_sw2=SW2; current_sw2<=SW2_max; current_sw2+=Step )); do
    TIMESTAMP=$(date +"%Y-%m-%d_%H-%M-%S")
    OutputFile="$DEC_DIR/${VIDEO_NAME}_LCEVC_${BaseEncoderUpper}_SW2-${current_sw2}_QP${QP}_${TIMESTAMP}.lvc"
   LogFile="$LOGS_DIR/${VIDEO_NAME}_LCEVC_${BaseEncoderUpper}_SW2-${current_sw2}_QP${QP}_${TIMESTAMP}.log"

```

```

echo -e "\n Rodando encode com SW2 = $current_sw2, QP = $QP" | tee -a "$STATUS_FILE"

${ModelEncoder} \
    --width=${WIDTH} --height=${HEIGHT} --format=${FORMAT} --fps=${FPS}
--qp=${QP} \
    --input_file="${InputFile}" \
    --output_file="${OutputFile}" \
    --base="${BIN_DIR}/${BASE_NAME}" \
    --base_encoder=${BaseEncoder} --encapsulation=nal \
    --cq_step_width_loq_0=${current_sw2} \
2>&1 | tee -a "${LogFile}"

DecodedFile="$DEC_DIR/${VIDEO_NAME}_Dec_${BaseEncoderUpper}_LCEVC_SW2-${current_sw2}_QP${QP}_${TIMESTAMP}.yuv"
DecodedLog="$LOGS_DIR/${VIDEO_NAME}_Dec_LCEVC_${BaseEncoderUpper}_SW2-${current_sw2}_QP${QP}_${TIMESTAMP}.log"

${ModelDecoder} \
    --base_encoder=${BaseEncoder} --fps=${FPS} \
    --input_file="${OutputFile}" \
    --output_file="${DecodedFile}" \
    --keep_base=true \
2>&1 | tee -a "${DecodedLog}"

base_file=$(find $SCRIPT_DIR -maxdepth 1 -type f -name "*base.es" | head -n 1)
enhance_file=$(find $SCRIPT_DIR -maxdepth 1 -type f -name "*enhancement.es" | head -n 1)

if [[ -f "$base_file" ]]; then
    base_size=$(stat -c%s "$base_file")
    # mv "$base_file" "$BASES_DIR/${VIDEO_NAME}_Base_LCEVC_${BaseEncoderUpper}_SW2-${current_sw2}_QP${QP}_${TIMESTAMP}.es"
    rm "$base_file"
else
    base_size=0
    echo "base.es nao encontrado!" | tee -a "$STATUS_FILE"
fi

```

```

if [[ -f "$enhance_file" ]]; then
    enhance_size=$(stat -c%s "$enhance_file")
    # mv "$enhance_file" "$BASES_DIR/${VIDEO_NAME}_Enha_LCEVC_${BaseEncoderUpper}_SW2-${current_sw2}_QP${QP}_${TIMESTAMP}.es"
    rm "$enhance_file"
else
    enhance_size=0
    echo "enhancement.es nao encontrado!" | tee -a "$STATUS_FILE"
fi

output_file_size=$(stat -c%s "$OutputFile")
if (( output_file_size > 0 )); then
    enhancement_ratio=$(awk "BEGIN { printf \"%.4f\", $enhance_size / $output_file_size * 100 }")
    bitrate_kbps=$(awk "BEGIN { printf \"%.2f\", ($output_file_size * 8 * $FPS) / ($NUM_FRAMES * 1000) }")
else
    enhancement_ratio=0
    bitrate_kbps=0
fi

echo -e "\n\tCalculando PSNR..." | tee -a "$STATUS_FILE"
$FFMPEG \
    -s:v "${WIDTH}x${HEIGHT}" -pix_fmt "${FORMAT}" -i "${InputFile}" \
    -s:v "${WIDTH}x${HEIGHT}" -pix_fmt "${FORMAT}" -i "${DecodedFile}" \
    -lavfi psnr="stats_file=${SCRIPT_DIR}/psnr.log" -f null - 2> /dev/null

psnr_avg=$(tail -n 1 ${SCRIPT_DIR}/psnr.log | awk -F'psnr_avg:' '{print $2}' | awk '{print $1}')
rm -f ${SCRIPT_DIR}/psnr.log >> "$STATUS_FILE"

if [[ -z "$psnr_avg" ]]; then
    psnr_avg="N/A"
fi

FileName="${OutputFile##*/}"
echo "${FileName},${current_sw2},${QP},${output_file_size},${base_size},${enhance_size},${enhancement_ratio}%,${psnr_avg},${bitrate_kbps}" | tee -a "$STATUS_FILE" >> "$SizesFile"

```

```

echo "${FileName},${current_sw2},${QP},${output_file_size},${base_size},${enhance_size},${enhancement_ratio}%,${psnr_avg},${bitrate_kbps}" >> "$LOCAL_CSV"

echo -e "\n\t SW2=${current_sw2} enhancement_ratio=${enhancement_ratio}% | psnr_avg=${psnr_avg} dB | bitrate=${bitrate_kbps} kbps\n" | tee -a "$STATUS_FILE"

rm -f $SCRIPT_DIR/*temp* 2>> "$STATUS_FILE"
done

echo -e "\n Script para QP=${QP} finalizado com sucesso!" | tee -a "$STATUS_FILE"

```

I.3 Resample

O script para *Resample* é o mesmo para os dois *mains*. O resultado do *Resample* foi desconsiderado por não contribuir de maneira útil para o resultado, mas ainda é necessário para que o script funcione.

resample_template.sh

```

#!/bin/bash

source ./config.sh

SCRIPT_PATH=$(readlink -f "$0")
SCRIPT_DIR=$(dirname "$SCRIPT_PATH")
STATUS_FILE="$SCRIPT_DIR/status.log"

ModelEncoder="/home/lucpena/apps/LTM/_build_linux/ModelEncoder"

rm $STATUS_FILE
echo -e "Iniciando Down+Up Sampling...\n" > "$STATUS_FILE"

SizesFileResample="$SCRIPT_DIR/${VIDEO_NAME}-Resample-Only.csv"

# Checa se o CSV ja existe, se nao, cria com o cabecalho
if [ ! -f "$SizesFileResample" ]; then

```

```

echo "file,sw2,qp,lvc_size,base_size,enhance_size,enhancement_ratio,
psnr_avg,bitrate_kbps" > "$SizesFileResample"
fi

FFMPEG="/home/lucpena/apps/FFMPEG-VMAF/ffmpeg"

Downsampled="$SCRIPT_DIR/${VIDEO_NAME}_downsampled.yuv"
Upsampled="$SCRIPT_DIR/${VIDEO_NAME}_upsampled.yuv"

echo -e "\nDownsampling...\n" | tee -a "$STATUS_FILE"
${ModelEncoder} --downsample_only=true --format=${FORMAT} -w ${WIDTH} -h ${HEIGHT} \
-i ${InputFile} -o ${Downsampled} >> "$STATUS_FILE"

echo -e "\nUpsampling...\n" | tee -a "$STATUS_FILE"
${ModelEncoder} --upsample_only=true --format=${FORMAT} -w ${WIDTH} -h ${HEIGHT} \
} \
-i ${Downsampled} -o ${Upsampled} >> "$STATUS_FILE"

echo -e "\nCalculando PSNR...\n" | tee -a "$STATUS_FILE"
$FFMPEG -s:v "${WIDTH}x${HEIGHT}" -pix_fmt ${FORMAT} -i ${InputFile} \
-s:v "${WIDTH}x${HEIGHT}" -pix_fmt ${FORMAT} -i ${Upsampled} \
-lavfi psnr="stats_file=psnr.log" -f null - >> "$STATUS_FILE"

psnr_avg=$(tail -n 1 psnr.log | awk -F'psnr_avg:' '{print $2}' | awk '{print $1}')
rm psnr.log >> "$STATUS_FILE"

output_file_size=$(stat -c%s "$Upsampled")
bitrate_kbps=$(awk "BEGIN { printf \"%.2f\", ($output_file_size * 8 * $FPS) / (
$NUM_FRAMES * 1000) }")

# Salva no CSV principal e no local
echo "${VIDEO_NAME}-Resample,0,0,$output_file_size,0,0,0%,${psnr_avg},0" >> "$SizesFile"
echo "${VIDEO_NAME}-Resample,0,0,$output_file_size,0,0,0%,${psnr_avg},0" >> "$SizesFileResample"

[[ -z "$psnr_avg" ]] && psnr_avg="N/A"
echo " Down-Up Sampling  PSNR=${psnr_avg} dB" >> "$STATUS_FILE"

```

```
echo -e "\n\n Downsampling + Upsampling finalizado!" | tee -a "$STATUS_FILE"  
rm $SCRIPT_DIR/*.yuv
```