

# Análise de Desempenho e Qualidade em Codificadores de Vídeo H.264 AVC e H.265 HEVC Rápidos e Embarcados

Amanda Szelag  
Universidade Federal do Pampa (UNIPAMPA)  
Bagé, RS, Brasil  
amandaszelag.aluno@unipampa.edu.br

Bernardo Rosa  
Instituto Federal Sul-rio-grandense (IFSul)  
Bagé, RS, Brasil  
bernardorosa.bg015@academico.ifsul.edu.br

Sara Vitória Henssler  
Universidade Federal do Pampa (UNIPAMPA)  
Bagé, RS, Brasil  
sarahenssler.aluno@unipampa.edu.br

Marcel Corrêa  
Instituto Federal Sul-rio-grandense (IFSul)  
Bagé, RS, Brasil  
marcelcorrea@ifsul.edu.br

## Abstract

Advanced Video Coding (H.264/AVC) and High Efficiency Video Coding (H.265/HEVC) are currently in a mature stage of their life cycle, with widespread hardware-accelerated encoding support across modern consumer devices. In contrast, the most recent standard, Versatile Video Coding (H.266/VVC), still lacks hardware-accelerated encoding support on general-purpose and graphics processors. This work evaluates the current state of hardware-accelerated AVC and HEVC encoding on modern Intel mobile CPUs using Intel Quick Sync Video (QSV). The results show that, although QSV-based encoders are generally capable of achieving real-time performance, this capability comes at the cost of a significant reduction in coding efficiency when compared to reference software implementations, with average BD-PSNR losses of approximately 4.6 dB for AVC and 4.1 dB for HEVC. Additionally, lower-tier CPUs were not always able to sustain real-time encoding for high-resolution video classes, requiring the use of faster presets that further degrade rate-distortion performance. The analysis also indicates that all evaluated processors rely on essentially the same hardware encoding architecture, with observed performance differences primarily attributed to variations in clock frequency, memory bandwidth, and overall CPU design constraints.

## Palavras-chave

Video Coding, HEVC, Hardware Acceleration

## 1 Introdução

Atualmente, as pessoas dependem cada vez mais de serviços digitais relacionados a vídeo para suas rotinas de lazer, estudo, trabalho e comunicação. Desta maneira, a codificação de vídeos digitais para fins de compressão torna-se indispensável, considerando que o volume de dados necessário para representar conteúdo visual sem compressão é impraticável.

Por exemplo, sem compressão, um filme com duração de 90 minutos, em resolução 1920x1080 *pixels*, atualização a 24 quadros por segundo (*frames per second* – fps), com 24 *bits* por *pixel*, teria um tamanho de aproximadamente 800 *gigabytes*. Este volume excede a capacidade total de armazenamento comumente encontrada em computadores e dispositivos móveis pessoais. Além disso, a transmissão em tempo real desse vídeo necessitaria de uma banda

aproximada de 1,2 Gbps, o que supera amplamente a banda de internet dos consumidores domésticos, bem como a banda de uma rede sem fio doméstica convencional.

No entanto, há décadas, existem ferramentas dedicadas à redução da quantidade de dados necessária para a reprodução de imagens e vídeos, tornando essa tarefa viável. Um *codec* (abreviação para codificador e decodificador), é uma ferramenta dedicada à comprimir e descomprimir vídeos, que pode ser implementada tanto em software puro quanto em hardware dedicado. *Codecs* fazem uso de conjunto amplo de algoritmos para fins de compressão. Contudo, em soluções que focam em codificação rápida e com baixo consumo de energia, seja por software ou hardware, muitos desses algoritmos precisam ser desabilitados ou bastante simplificados, resultando em uma perda na qualidade de compressão em comparação com codificadores que implementam mais dos algoritmos disponíveis.

O padrão de codificação de vídeo H.264 *Advanced Video Coding* (AVC) [1] foi desenvolvido em conjunto pelos grupos *Video Coding Experts Group* (VCEG) e *Moving Picture Experts Group* (MPEG), sendo finalizado e publicado em 2003 como um sucessor do padrão H.262/MPEG-2 [2]. O padrão de codificação de vídeo H.265 *High Efficiency Video Coding* (HEVC) [3] também foi desenvolvido em conjunto pelos grupos VCEG e MPEG, sendo finalizado e publicado em 2013, trazendo até 50% de redução na taxa de *bits* para uma mesma qualidade perceptiva em relação ao predecessor [4]. Padrões como o HEVC, o AVC e o MPEG-2 [2] foram peças fundamentais na popularização de *streaming*, transmissão de TV digital e mídias físicas digitais para reprodução de vídeo, tornando vídeos digitais um dos principais pilares da internet nos dias de hoje.

O lançamento do H.266 Versatile Video Coding (VVC) [5, 6] e do AOMedia Video 1 (AV1) [7, 8] estabeleceu esses dois formatos como o estado da arte em codificação de vídeo. Apesar disso, tanto o AVC quanto o HEVC encontram-se atualmente em uma fase de seu ciclo de mercado em que o suporte à aceleração por hardware está amplamente difundido em dispositivos atuais voltados ao consumo de vídeo, como TVs, *smartphones*, computadores portáteis, consoles de videogame e outros. Em contraste, o suporte ao VVC e AV1 ainda é, em geral, ausente para consumidores ou restrito a dispositivos de altíssimo custo, porque o desenvolvimento de aceleração por hardware para novos formatos e adoção pelos fabricantes é um processo custoso e bastante demorado.

Assim, o objetivo desse estudo é avaliar o estado atual da aceleração por hardware para codificadores AVC e HEVC em dispositivos

modernos em termos de tempo de execução, taxa de compressão e fidelidade visual resultante. Para a realização da análise, foram selecionados três processadores *Intel* da mesma geração e da linha *mobile*, representando diferentes segmentos de mercado: um modelo de entrada, um intermediário e um de alto desempenho.

Esse artigo está organizado da seguinte forma: a Seção 2 apresenta brevemente fundamentos sobre codificação de vídeo. A Seção 3 detalha a metodologia adotada. A Seção 4 apresenta e discute os resultados dos experimentos. Por fim, a Seção 5 conclui esse artigo.

## 2 Fundamentos de Codificação de Vídeo

A maioria dos codificadores modernos baseia-se nas seguintes operações de processamento de sinais e dados: (i) particionamento de blocos, (ii) predição intra-quadro e inter-quadros, (iii) transformadas (módulo T), (iv) quantização (módulo Q) e (v) codificação de entropia, conforme ilustrado na Figura 1.

Além disso, codificadores possuem também um caminho de reconstrução (loop de decodificação), ilustrado pela linha tracejada na Figura 1, contendo a quantização inversa (módulo IQ) e as transformas inversas (módulo IT). Essa característica de projeto garante que o codificador dependa exclusivamente de quadros de referência acessíveis ao decodificador, possibilitando que as mesmas predições sejam feitas por ambos. Adicionalmente, um módulo opcional de filtragem de laço pode ser incluído ao final do caminho de reconstrução para melhorar a qualidade visual subjetiva dos quadros reconstruídos.

O particionamento de blocos consiste na divisão do quadro em unidades menores, que podem ser processadas de forma independente para fins de compressão. Essa estratégia permite maior adaptabilidade às características espaciais e temporais da imagem, viabilizando uma codificação mais eficiente. Após o particionamento, o codificador realiza a predição de cada bloco com base em informações disponíveis. Na predição intra-quadro, os blocos são preditos com base em amostras já codificadas do próprio quadro atual, explorando correlações espaciais. Já na predição inter-quadros, são utilizados blocos de quadros já codificados referência, explorando a redundância temporal entre quadros vizinhos. Em ambos os casos, o resultado da predição é subtraído do bloco original, produzindo um resíduo (diferença entre o sinal original e o predito). A eficiência das técnicas de predição tem impacto direto na magnitude do resíduo e na eficiência da compressão [9].

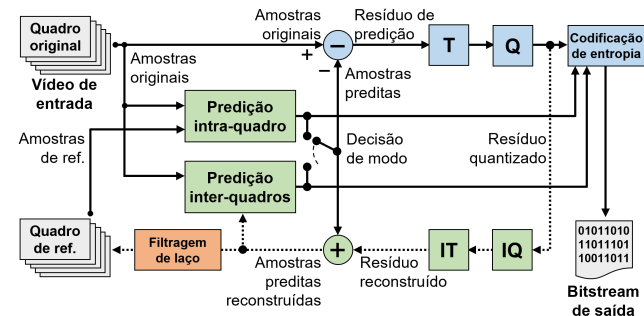


Figura 1: Diagrama de blocos de um codificador de vídeo genérico.

O resíduo gerado após a predição é submetido a uma transformada, com o objetivo de concentrar a energia do sinal em um número reduzido de coeficientes de baixas frequências espaciais, facilitando a compressão posterior. Nessa etapa, em padrões mais modernos, pode ocorrer um novo particionamento, no qual os blocos residuais são divididos em blocos de transformada com tamanhos menores, adaptando-se melhor às características locais do conteúdo [9]. Enquanto padrões mais antigos costumam se basear apenas na Transformada Discreta dos Cossenos (*Discrete Cosine Transform - DCT*), padrões mais modernos permitem um conjunto ampliado de transformadas [10].

Em seguida, os coeficientes transformados são quantizados, reduzindo a precisão dos valores representados com o intuito de eliminar informações menos relevantes perceptualmente. O grau de perdas irreversíveis introduzido nessa etapa é controlado pelo parâmetro de quantização (*Quantization Parameter - QP*): quanto maior o valor do QP, maior a perda de informação e maior a taxa de compressão resultante, e vice-versa [10]. Esta etapa define o conceito de compressão com perdas (*lossy compression*).

Por fim, os coeficientes quantizados, juntamente com os parâmetros de codificação utilizados, como modos de predição, vetores de movimento, tamanhos de blocos e demais sinais, são encaminhados à etapa de codificação de entropia. Essa etapa aplica algoritmos de compressão sem perdas, que visam reduzir a redundância estatística dos dados codificados, atribuindo representações mais curtas para símbolos mais frequentes. O resultado final é um *bitstream* comprimido, estruturado de acordo com as especificações do padrão de codificação vídeo, que pode então ser armazenado ou transmitido para posterior decodificação e reprodução [9].

Além das etapas descritas, codificadores modernos utilizam mecanismos de controle baseados em *Rate-Distortion Optimization (RDO)* [11] para selecionar, entre múltiplas opções de codificação possíveis, aquela que oferece o melhor compromisso entre taxa de bits e distorção. Nesse processo, diferentes decisões, como modos de predição, tamanhos de blocos, vetores de movimento, estruturas de partição e escolhas de transformada, são avaliadas por meio de uma função de custo que combina a distorção introduzida e a taxa de bits necessária para sinalizar cada alternativa. A utilização extensiva do RDO é um dos principais fatores responsáveis pela elevada eficiência de codificação observada em implementações de referência, ao custo de um aumento significativo da complexidade computacional, decorrente da exploração de um amplo espaço combinatorial de decisões de codificação.

## 3 Metodologia de Pesquisa

Neste estudo foram avaliados os seguintes codificadores:

- **Joint Test Model (JM) v19.1** [12]: Software mantido pelo grupo responsável pelo padrão JM que serve como implementação de referência, com objetivo de garantir a conformidade com as especificações e demonstrar potencial máximo de compressão;
- **HEVC Test Model (HM) v18.0** [13]: Similar ao JM, mas para o padrão HEVC;
- **h264\_qsv** [14]: Implementação do codificador AVC integrada ao FFmpeg, que utiliza a tecnologia *Intel Quick Sync Video (QSV)*. Diferente de bibliotecas como o x264 [15],

que realizam a compressão em software puro, o h264\_qsv aproveita diretamente o hardware dedicado nas unidades de processamento gráfico (*Graphics Processing Unit* - GPU) integradas aos processadores *Intel*, realizando a codificação de forma puramente em hardware;

- **hevc\_qsv** [14]: Similar ao h264\_qsv, mas para o padrão HEVC.

### 3.1 Métricas Utilizadas

O desempenho de cada codificador foi medido em termos de tempo de codificação, taxa de *bits* e distorção, esta última quantificada por meio da métrica *Peak Signal-to-Noise Ratio* (PSNR).

A taxa de *bits* (*bitrate*) é uma métrica que indica a quantidade de dados transmitidos ou processados por segundo em um arquivo de vídeo. Ela é geralmente expressa em *kilobits* por segundo (kbps). Um *bitrate* elevado pode sugerir maior fidelidade visual, mas não necessariamente, pois também pode ser resultado de uma compressão ineficiente, da mesma forma que um *bitrate* reduzido não implica obrigatoriamente em baixa qualidade, podendo indicar um processo de compressão mais eficiente.

A distorção em um vídeo refere-se às diferenças introduzidas entre o sinal original e o sinal comprimido, resultando em perda de qualidade ou fidelidade visual. Essas diferenças podem se manifestar como borrões, blocos, ruídos ou perda de detalhes finos na imagem. Para medir esse erro, é comum o uso de métricas objetivas que comparam o vídeo reconstruído com o vídeo de referência. Uma das métricas mais utilizadas é o *Peak Signal-to-Noise Ratio* (PSNR), que se baseia no cálculo do *Mean Squared Error* (MSE) entre os dois sinais. O MSE mede a média dos erros quadráticos entre os *pixels* correspondentes, enquanto o PSNR expressa essa relação em decibéis (dB), fornecendo uma indicação da qualidade percebida: valores mais altos de PSNR indicam menor distorção e maior similaridade com o original.

As equações (1) e (2) demonstram o cálculo do MSE e PSNR, respectivamente. Nelas,  $O(i, j)$  representa o pixel da imagem ou vídeo original,  $T(i, j)$  o pixel correspondente na versão de teste (comprimida),  $M$  e  $N$  indicam as dimensões da imagem (largura e altura), e  $MAX$  corresponde ao valor máximo possível de intensidade de *pixel*, como 255 no caso de imagens de 8 *bits* [9].

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [O(i, j) - T(i, j)]^2 \quad (1)$$

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

Ao se obter resultados de *bitrate* e PSNR em pelo menos quatro pontos de operação de um codificador, é possível plotar estes pontos em um gráfico de taxa-distorção (*Rate-distortion* - RD), como ilustra a Figura 2. Esses pontos permitem a visualização do *bitrate* utilizado com a distorção introduzida para cada ponto de operação. No entanto, a comparação entre dois codificadores não é trivial se considerarmos apenas uma métrica isolada. Por exemplo, o codificador A pode apresentar menor *bitrate* para um mesmo conteúdo, mas ao custo de maior distorção (menor PSNR) em relação ao codificador B. Nesse caso, surge a dúvida: qual dos dois é realmente melhor?

Essa dificuldade motiva o uso de métricas comparativas mais robustas, capazes de comparar o desempenho global dos codificadores ao longo da curva de taxa-distorção. É nesse contexto que se inserem medidas como *Bjontegaard Delta Bitrate* (BD-BR) e *Bjontegaard Delta PSNR* (BD-PSNR) [17]. A ideia central é que os pontos de operação obtidos para cada codificador (valores de *bitrate* e PSNR em diferentes configurações) são usados para gerar curvas de taxa-distorção por meio de interpolação.

Uma vez construídas as curvas RD, procede-se ao cálculo da diferença média entre elas ao longo do intervalo comum de operação. No caso do BD-BR, integra-se a diferença do *bitrate* em escala logarítmica como função do PSNR, resultando na variação percentual média de taxa necessária para que um codificador atinja a mesma qualidade do outro. De forma análoga, o BD-PSNR é obtido pela integração da diferença de PSNR como função do *bitrate* em escala logarítmica, expressando o ganho ou a perda média de qualidade, em decibéis, para uma mesma taxa de *bits*. A Figura 3 ilustra este procedimento para o cálculo do BD-BR. O valor final da métrica corresponde à média dessa área integrada ao longo do intervalo considerado.

### 3.2 Experimentos Realizados

**3.2.1 Codificações de Referência.** A primeira etapa dos experimentos consistiu na geração dos resultados de referência utilizando o JM v19.1 [12] e HM v18.0 [13]. É importante reforçar que implementações de referência não são voltadas para uso prático, pois empregam um conjunto extenso de técnicas de codificação que, embora melhorem muito os resultados do codificador, também tornam o tempo de codificação extremamente elevado. Portanto, devido ao alto esforço computacional associado ao JM e HM, os experimentos foram executados em um servidor de alto desempenho equipado com dois processadores *Intel Xeon Gold 6430* (2,1 GHz), totalizando 128 núcleos lógicos, e quatro módulos de memória DDR5 (4.400 MHz), totalizando 128 GB.

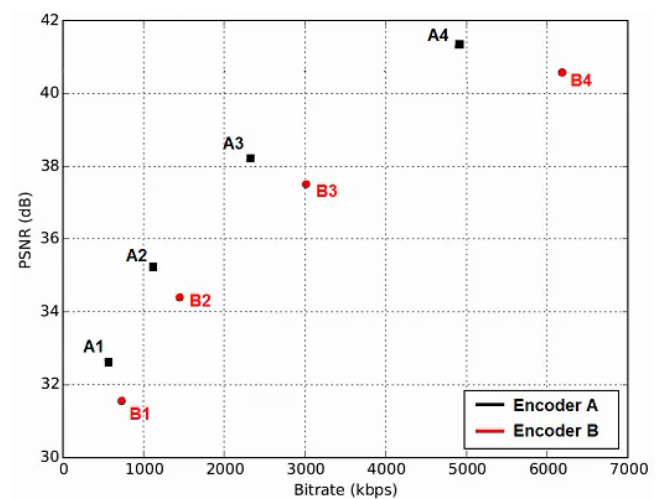


Figura 2: Gráfico de taxa-distorção entre dois codificadores hipotéticos A e B [16]

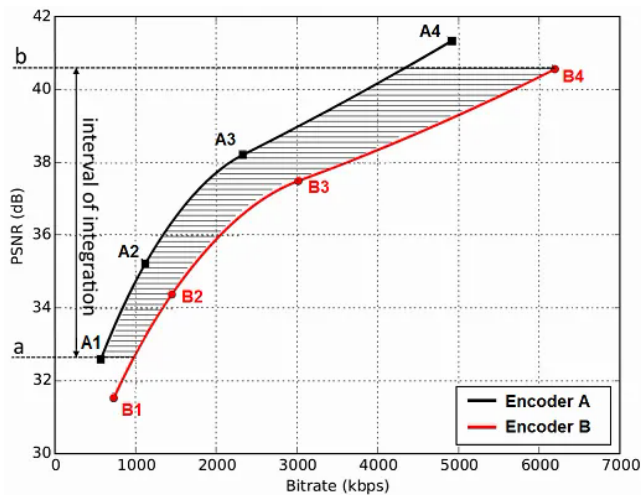


Figura 3: Gráfico com curvas de taxa-distorção interpoladas e integração que representa o BD-BR entre os codificadores [16].

Nesse estudo, foram realizados experimentos de codificação de vídeo seguindo recomendações presentes no documento que define condições comuns de teste (*Common Test Conditions – CTCs*) [18]. Entre outras orientações, o documento estabelece um conjunto padrão de vídeos de teste e parâmetros de codificação a serem utilizados com o JM e HM. Além disso, sugere que cada vídeo seja codificado com o JM e HM quatro vezes, considerando o conjunto de QPs [22, 27, 32, 37] (pontos de operação). Essas diretrizes são fundamentais para garantir a comparabilidade e a reprodutibilidade dos resultados entre diferentes estudos.

Os vídeos utilizados abrangem as classes B a F das CTCs, totalizando 20 vídeos de teste com resoluções e características variadas. Os vídeos da classe B possuem resolução 1080p (1920x1080 *pixels*), os da classe C em 480p (832x480 *pixels*), os da classe D em 240p (416x240 *pixels*), enquanto os da classe E apresentam resolução 720p (1280x720 *pixels*) e natureza quase estática. Por fim, a classe F inclui vídeos com resoluções variadas, voltados para o *Screen Content Coding* (SCC), contendo elementos típicos de telas, como textos, gráficos e interfaces [18].

**3.2.2 Codificação com Hardware Dedicado (QSV).** A segunda etapa consistiu em gerar os resultados de codificação com hardware dedicado, utilizando o software FFmpeg v6.1.1 [19] como interface com as bibliotecas qsv\_h264 e qsv\_hevc. Com o objetivo de viabilizar a análise por meio da métrica BD-BR, que é mais compreendida por leigos por expressar a variação média de *bitrate* para uma mesma qualidade, a metodologia adotada inicialmente priorizou o alinhamento das curvas de taxa-distorção para que tenham pontos em comum no eixo da métrica de distorção, tal como ilustra a Figura 3. Para isso, os codificadores QSV foram configurados em um modo de qualidade constante, com a qualidade alvo sendo ajustada manualmente para cada sequência através do parâmetro *global\_quality* de forma a se obter níveis de PSNR próximos aos produzidos pelos softwares de referência.

Entretanto, verificou-se que essa metodologia não era adequada, uma vez que, mesmo para os parâmetros de qualidade mais altos, as codificações com o QSV não produziram curvas de taxa-distorção com valores de PSNR em comum com aquelas obtidas pelos softwares de referência. Em outras palavras, a maior fidelidade visual alcançada pelo QSV não se mostrou comparável à obtida pelas implementações de referência, mesmo para o QP 37 (mais agressivo). Como consequência, não foi possível estabelecer uma região comum de qualidade, tornando a aplicação da métrica BD-BR matematicamente inviável.

Diante disso, a metodologia final adotada consistiu na configuração do qsv\_h264 e qsv\_hevc em modo de controle de taxa por *bitrate* alvo (*Variable Bitrate – VBR*), de forma que cada codificação busque atingir o mesmo *bitrate* obtido pelo JM e HM em suas codificações com os valores de QP 22, 27, 32 e 37. Assim, as curvas resultantes passam a ter intervalos comuns no eixo do *bitrate*, sendo possível calcular a variação de PSNR para uma mesma qualidade através do BD-PSNR.

Os experimentos foram executados em três processadores *mobile* de segmentos distintos, mas da mesma geração: *Intel Core i3-1305U*, *Intel Core i5-1335U* e *Intel Core i7-13650HX*. A escolha por processadores de mesmo fabricante, geração e com projetos térmicos (*Thermal Design Power – TDP*) semelhantes tem como objetivo garantir uma comparação justa entre diferentes níveis de capacidade computacional, minimizando a interferência de variáveis como diferenças arquiteturais ou grandes discrepâncias de TDP observadas principalmente entre plataformas *desktop* e *mobile*.

Vale destacar que o QSV oferece *presets* para estes processadores, sendo que os mais rápidos desabilitam ou simplificam diversas técnicas de compressão, obtendo maior velocidade à custa de degradação na taxa-distorção. Os três *presets* utilizados nos experimentos foram *veryslow*, *medium* e *veryfast*. Todos os testes foram inicialmente executados com o *preset veryslow* (mais lento e com melhores resultados de qualidade), sendo ajustados para um *preset* mais rápido apenas nos casos em que o tempo de execução ficou abaixo do tempo real. Para fins de reprodutibilidade do trabalho, apenas os parâmetros de *bitrate* alvo e *preset* foram alterados, sendo os demais parâmetros, como números de quadros de referência, uso de B-frames, tamanho do *Group of Pictures* (GOP), mantidos *default*.

## 4 Resultados e Discussão

As Tabelas 1 e 2 apresentam os resultados obtidos para os cenários de codificação AVC e HEVC, respectivamente, incluindo os tempos de execução dos softwares de referência e os resultados de taxa-distorção e desempenho temporal das implementações com aceleração por hardware. Os valores são reportados como médias por classe de vídeo, conforme definido nas CTCs [18]. A eficiência de codificação é quantificada por meio da métrica BD-PSNR [17], que expressa a variação média de PSNR para uma mesma taxa de bits, sendo valores negativos indicativos de perda de qualidade. O desempenho temporal é apresentado em relação ao tempo real (1×), de modo que valores superiores a 1× indicam processamento em tempo real.

Inicialmente, observa-se que os softwares de referência apresentam tempos de execução extremamente inferiores ao tempo real. Em média, o JM demanda aproximadamente 192 vezes a duração do

Tabela 1: Resultados de tempo de execução para o software de referência do AVC e resultados de taxa-distorção e tempo para os dispositivos usando aceleração h264\_qsv.

Classe (Média)	JM v19.1	Core i3-1305U		Core i5-1335U		Core i7-13650HX	
	Tempo	BD-PSNR	Tempo	BD-PSNR	Tempo	BD-PSNR	Tempo
B	0,0011×	-4,2472 dB	1,01×	-4,2372 dB	1,69×	-4,1959 dB	2,47×
C	0,0038×	-7,0648 dB	4,31×	-7,0648 dB	4,72×	-7,0648 dB	6,21×
D	0,0153×	-3,9985 dB	11,93×	-3,9985 dB	11,55×	-3,9985 dB	14,39×
E	0,0015×	-2,7194 dB	1,57×	-2,5881 dB	2,28×	-2,5881 dB	3,22×
F	0,0046×	-4,8074 dB	6,93×	-4,8074 dB	7,67×	-4,8074 dB	9,88×
Geral	0,0052×	-4,6438 dB	5,15×	-4,6216 dB	5,58×	-4,6113 dB	7,23×

Tabela 2: Resultados de tempo de execução para o software de referência do HEVC e resultados de taxa-distorção e tempo para os dispositivos usando aceleração hevc\_qsv.

Classe (Média)	HM v18.0	Core i3-1305U		Core i5-1335U		Core i7-13650HX	
	Tempo	BD-PSNR	Tempo	BD-PSNR	Tempo	BD-PSNR	Tempo
B	0,00016×	-3,3194 dB	1,02×	-3,3044 dB	1,09×	-3,2587 dB	1,97×
C	0,00055×	-5,9872 dB	2,79×	-5,9872 dB	2,47×	-5,9872 dB	2,99×
D	0,00267×	-3,2052 dB	5,15×	-3,2052 dB	4,68×	-3,2052 dB	5,16×
E	0,00032×	-2,6063 dB	1,78×	-2,4637 dB	1,64×	-2,4637 dB	2,51×
F	0,00064×	-5,4690 dB	4,03×	-5,4690 dB	3,88×	-5,4690 dB	4,63×
Geral	0,00087×	-4,1531 dB	2,96×	-4,1279 dB	2,75×	-4,1165 dB	3,45×

vídeo para completar a codificação, enquanto o HM, refletindo o aumento substancial de complexidade introduzido pelo padrão HEVC, requer cerca de 1149 vezes a duração do conteúdo. Esses resultados reforçam o papel exclusivo dessas implementações como ferramentas de pesquisa e validação normativa, não sendo adequadas para aplicações práticas de codificação em tempo real.

Em contraste, as implementações baseadas em QSV atingem, em média, processamento em tempo real nos três processadores avaliados. Entretanto, essa capacidade está fortemente condicionada à resolução, à classe de vídeo e ao *preset* adotado. Nos processadores Core i3-1305U e Core i5-1335U, particularmente para a classe B (1080p), foi necessário o uso de *presets* mais rápidos, o que, em alguns casos, ainda não foi suficiente para garantir processamento em tempo real em todos os vídeos. Esse comportamento evidencia que, mesmo com aceleração por hardware dedicado, a codificação em tempo real não é garantida em cenários de maior complexidade, especialmente em dispositivos de menor capacidade computacional.

Do ponto de vista da eficiência de codificação, observa-se uma perda expressiva de qualidade para uma mesma taxa de bits quando comparadas as implementações QSV aos respectivos softwares de referência. Para o h264\_qsv, a perda média de BD-PSNR foi de aproximadamente -4,6 dB, enquanto para o hevc\_qsv esse valor ficou em torno de -4,1 dB. Considerando a natureza logarítmica do PSNR, conforme mostra a equação (2), tais reduções representam uma degradação muito significativa da fidelidade visual, indicando que o ganho de desempenho temporal proporcionado pelo hardware dedicado ocorre à custa de uma redução substancial na eficiência de compressão.

Essas perdas não podem ser atribuídas apenas a ajustes de *preset*, mas refletem limitações estruturais relacionadas ao projeto de codificadores em hardware *fixed-function*. Ao contrário dos softwares de referência, que exploram extensivamente técnicas de otimização baseadas em RDO [11] e avaliam um amplo espaço de modos de codificação, as implementações em hardware priorizam determinismo, baixa latência e paralelismo previsível. Como consequência, diversas ferramentas avançadas de compressão são simplificadas ou parcialmente desabilitadas, resultando em perdas sistemáticas de eficiência.

Um aspecto relevante observado nos resultados é que a diferença média de eficiência entre os codificadores de referência e o QSV é ligeiramente menor no HEVC do que no AVC. Esse comportamento sugere que parte do aumento de complexidade introduzido pelo padrão HEVC não é plenamente explorada pelas implementações em hardware, fazendo com que o ganho teórico do *codec* de até 50% [4] seja apenas parcialmente refletido no cenário prático de codificação acelerada. Tal resultado indica que a distância entre o potencial máximo de compressão e as soluções práticas tende a se manter, ou até a se ampliar, à medida que novos padrões incorporam ferramentas cada vez mais complexas.

A análise por classe de vídeo também revela dependência significativa em relação às características do conteúdo. Perdas mais acentuadas são observadas em classes com maior riqueza de textura, movimento complexo ou conteúdo artificial, como as classes C e F, enquanto conteúdos quase estáticos, como os da classe E, apresentam degradações mais moderadas. Esse comportamento indica que o QSV é mais eficiente em cenários com maior previsibilidade

espacial e temporal, apresentando limitações mais evidentes em conteúdos desafiadores do ponto de vista de compressão.

Por fim, observa-se que, independentemente do segmento do processador, os resultados de taxa-distorção são idênticos nas classes em que o mesmo *preset* pôde ser mantido. Isso confirma que os três processadores utilizam essencialmente a mesma arquitetura embarcada de codificação por hardware, sendo as diferenças de desempenho atribuídas a fatores como frequência de operação, largura de banda de memória e características gerais da plataforma, e não ao bloco de codificação de vídeo em si.

## 5 Conclusão

Este trabalho avaliou o desempenho de codificadores de vídeo AVC e HEVC com aceleração por hardware em processadores móveis modernos da *Intel*, comparando-os com os respectivos softwares de referência. Os resultados demonstraram que a codificação em tempo real é viável em dispositivos móveis por meio do uso de hardware dedicado, porém essa viabilidade é alcançada à custa de uma redução expressiva na eficiência de compressão.

As perdas observadas em termos de BD-PSNR evidenciam que as implementações baseadas em QSV operam em um regime de qualidade significativamente distinto daquele alcançado pelos softwares de referência. Tal comportamento decorre de escolhas arquiteturais inerentes a codificadores *fixed-function*, que priorizam determinismo, previsibilidade e baixa latência em detrimento da exploração completa das ferramentas avançadas de compressão previstas nos padrões. Essas limitações tornam-se particularmente evidentes em conteúdos de maior complexidade espacial ou temporal, nos quais a degradação visual tende a ser mais perceptível.

Além disso, os resultados indicam que, embora os processadores avaliados pertençam a segmentos distintos, todos compartilham essencialmente a mesma arquitetura de codificação por hardware, de modo que as diferenças observadas estão relacionadas principalmente à capacidade de sustentar o processamento em tempo real sob diferentes condições de carga. Processadores de entrada e intermediários apresentaram maior dificuldade em manter esse requisito em cenários mais exigentes, exigindo ajustes de velocidade que implicam degradação adicional da qualidade.

De forma geral, os resultados reforçam a existência de uma lacuna significativa entre o potencial máximo de compressão representado por implementações de referência e as soluções práticas de codificação acelerada por hardware. Essa lacuna tende a se tornar ainda mais relevante com a adoção de padrões mais recentes e complexos, como AV1 e VVC, destacando a importância de pesquisas futuras voltadas ao projeto de hardware e software.

Para trabalhos futuros, pretende-se incorporar métricas perceptuais mais avançadas, como o *Structural Similarity Index* (SSIM) [20] e, principalmente, o *Video Multimethod Assessment Fusion* (VMAF) [21]. O VMAF tem se consolidado como uma métrica de referência na indústria e na comunidade acadêmica por apresentar maior correlação a percepção visual humana ao combinar múltiplos indicadores de qualidade estrutural, perceptual e temporal. A adoção dessas métricas permitirá uma análise mais abrangente do impacto das perdas de eficiência observadas, especialmente em cenários

nos quais diferenças moderadas de PSNR podem resultar em degradações visuais significativamente distintas do ponto de vista do usuário.

## Agradecimentos

Este trabalho foi financiado pelo Instituto Federal Sul-rio-grandense (IFSul) e pela Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul (FAPERGS) – Projeto 25/2551-0000903-0.

## Referências

- [1] ITU-T. Advanced video coding for generic audiovisual services. <https://www.itu.int/rec/t-rec-h.264>, Aug. 2024. Recommendation ITU-T H.264 (V15) (08/2024).
- [2] ITU-T. Generic coding of moving pictures and associated audio information. <https://www.itu.int/rec/t-rec-h.262>, Feb. 2012. Recommendation ITU-T H.262 (02/2012).
- [3] ITU-T. High Efficiency Video Coding. <https://www.itu.int/rec/t-rec-h.265>, Jul. 2024. Recommendation ITU-T H.265 (V10) (07/2024).
- [4] Gary J. Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12):1649–1668, 2012. doi: 10.1109/TCSVT.2012.2221191.
- [5] ITU-T. Versatile Video Coding. <https://www.itu.int/rec/t-rec-h.266>, Sep. 2023. Recommendation ITU-T H.266 (V3) (09/2023).
- [6] Benjamin Bross, Ye-Kui Wang, Yan Ye, Shan Liu, Jianle Chen, Gary J. Sullivan, and Jens-Rainer Ohm. Overview of the versatile video coding (vvc) standard and its applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10):3736–3764, 2021. doi: 10.1109/TCSVT.2021.3101953.
- [7] AOMedia. AV1 Bitstream & Decoding Process Specification (v1.0.0-errata1). <https://github.com/AOMediaCodec/av1-spec/releases/tag/v1.0.0-errata1>, Jan. 2019.
- [8] Jingning Han, Bohan Li, Debargha Mukherjee, Ching-Han Chiang, Adrian Grange, Cheng Chen, Hui Su, Sarah Parker, Sai Deng, Urvang Joshi, Yue Chen, Yunqing Wang, Paul Wilkins, Yaowu Xu, and James Bankoski. A technical overview of av1. *Proceedings of the IEEE*, 109(9):1435–1462, 2021. doi: 10.1109/JPROC.2021.3058584.
- [9] I.E. Richardson. *Coding Video: A Practical Guide to HEVC and Beyond*. Wiley, 2024. ISBN 9781118711781.
- [10] V. Sze, M. Budagavi, and G.J. Sullivan. *High Efficiency Video Coding (HEVC): Algorithms and Architectures*. Integrated Circuits and Systems. Springer International Publishing, 2014. ISBN 9783319068954.
- [11] G.J. Sullivan and T. Wiegand. Rate-distortion optimization for video compression. *IEEE Signal Processing Magazine*, 15(6):74–90, 1998. doi: 10.1109/79.733497.
- [12] JVET. Jm, 2025. URL <https://vcgit.hhi.fraunhofer.de/jvet/JM>. [Online].
- [13] JVET. Hm, 2025. URL <https://vcgit.hhi.fraunhofer.de/jvet/HM>. [Online].
- [14] Intel Corporation. Intel video processing library. <https://github.com/intel/libvpl>, 2025. [Online].
- [15] VideoLAN. libx264. <https://code.videolan.org/videolan/x264/>, 2025. [Online].
- [16] M. Sharabayko. Bjontegaard delta-rate metric. <https://medium.com/innovation-labs-blog/bjontegaard-delta-rate-metric-c8c82c1bc42c>, 2023. [Online].
- [17] G. Bjontegaard. Calculation of average psnr differences between rd-curves. VCEG document VCEG-M33, March 2001. URL [https://www.itu.int/wftp3/av-arch/video-site/0104\\_Aus/VCEG-M33.doc](https://www.itu.int/wftp3/av-arch/video-site/0104_Aus/VCEG-M33.doc). [Online].
- [18] K. Sharman and K. Sühring. Common test conditions for hm video coding experiments. JCT-VC document JCTVC-AC1100, January 2018. URL [http://phenix.int-evry.fr/jct/doc\\_end\\_user/documents/29\\_Macau/wg11/JCTVC-AC1100-v1.zip](http://phenix.int-evry.fr/jct/doc_end_user/documents/29_Macau/wg11/JCTVC-AC1100-v1.zip). [Online].
- [19] Ffmpeg. Ffmpeg. <https://ffmpeg.org/>, 2024. [Online].
- [20] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861.
- [21] et al. Z. Li. Toward a practical perceptual video quality metric, 2016. URL <https://netflixtechblog.com/toward-a-practical-perceptual-video-quality-metric-653f208b9652>. [Online].