



PROJETO FINAL DE GRADUAÇÃO

**ESTUDO DA QUALIDADE DE TRANSCODIFICAÇÃO
DO SOFTWARE FFMPEG**

Julianna Regina Brandão Nunes
Victor Nunes Gomes

Brasília, Julho de 2018

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

PROJETO FINAL DE GRADUAÇÃO

**ESTUDO DA QUALIDADE DE TRANSCODIFICAÇÃO
DO SOFTWARE FFMPEG**

Julianna Regina Brandão Nunes
Victor Nunes Gomes

*Projeto Final de Graduação submetido ao Departamento de Engenharia
Elétrica como requisito parcial para obtenção
do grau de Engenheiro de Redes de Comunicação.*

Banca Examinadora

Prof. Eduardo Peixoto Fernandes da Silva, _____
UnB-ENE
Orientador

Prof. Daniel Guerreiro e Silva, UnB-ENE _____
Examinador interno

Prof. Camilo Chang Dorea, UnB-CIC _____
Examinador interno

FICHA CATALOGRÁFICA

NUNES, JULIANNA REGINA BRANDÃO ; GOMES, VICTOR NUNES
ESTUDO DA QUALIDADE DE TRANSCODIFICAÇÃO DO SOFTWARE FFMPEG [Distrito Federal] 2018.

xvi, 31 p., 210 x 297 mm (ENE/FT/UnB, Engenheiro, Engenharia Elétrica, 2018).

Projeto Final de Graduação - Universidade de Brasília, Faculdade de Tecnologia.

Departamento de Engenharia Elétrica

1. Transcodificação

2. Métricas de Qualidade

3. FFMPEG

4. Vídeo

I. ENE/FT/UnB

II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

NUNES, J. R. B. ; GOMES, V. N. (2018). *ESTUDO DA QUALIDADE DE TRANSCODIFICAÇÃO DO SOFTWARE FFMPEG*. Projeto Final de Graduação, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 31 p.

CESSÃO DE DIREITOS

AUTORES: Julianna Regina Brandão Nunes, Victor Nunes Gomes

TÍTULO: ESTUDO DA QUALIDADE DE TRANSCODIFICAÇÃO DO SOFTWARE FFMPEG.

GRAU: Engenheiro de Redes de Comunicação

ANO: 2018

É concedida à Universidade de Brasília permissão para reproduzir cópias deste Projeto Final de Graduação e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. Os autores reservam outros direitos de publicação e nenhuma parte desse Projeto Final de Graduação pode ser reproduzida sem autorização por escrito dos autores.

Julianna Regina Brandão Nunes

Victor Nunes Gomes

Depto. de Engenharia Elétrica (ENE) - FT

Universidade de Brasília (UnB)

Campus Darcy Ribeiro

CEP 70919-970 - Brasília - DF - Brasil

DEDICATÓRIAS

Dedico este trabalho às verdadeiras pessoas de Jack Kerouac: aos loucos para viver, para falar, para serem salvos, que querem tudo ao mesmo tempo agora, aqueles que nunca bocejam, mas queimam como fabulosos fogos de artifício explodindo como constelações

Juliana Regina Brandão Nunes

Dedico este trabalho à minha família, à minha namorada e aos meus amigos, os quais tiveram importante contribuição para que eu pudesse concluir o curso de graduação.

Victor Nunes Gomes

AGRADECIMENTOS

Agradeço à minha família, aos meus amigos nessa ponte aérea São Paulo-Brasília-Bahia e principalmente a cada um que apareceu no meu caminho nesses anos todos, que fizeram essa minha trajetória ser tão única e especial. Agradecimento especial também ao Victor por todo auxílio, dedicação e compreensão e ao professor Eduardo Peixoto pela ótima orientação.

Juliana Regina Brandão Nunes

Agradeço primeiramente à Deus pelo dom da vida, à minha família pelo apoio durante todos esses anos de estudo, à minha namorada Yasmin pelo companherismo e apoio durante a conclusão do curso, aos meus amigos e pessoas que me ajudaram crescer durante minha graduação tanto pessoalmente como profissionalmente. Agradeço também à Juliana por todo o esforço depositado em nosso trabalho. Por fim agradeço ao professor Eduardo Peixoto pelo apoio, paciência e dedicação dados para a realização desse projeto

Victor Nunes Gomes

Em um ambiente de emissora de televisão, diariamente são recebidos centenas de vídeos gravados por dispositivos diferentes e em diversos formatos, os quais, na maioria dos casos, não estão no padrão desejado para que a transmissão possa ser realizada.

A solução técnica que permite a transmissão de vídeo se chama transcodificação de vídeo. A transcodificação é o processo que consiste na conversão do formato do vídeo, converte um fluxo de bits compactado (chamado fluxo de bits de origem ou entrada) para outro fluxo de bits compactado (chamado de fluxo de bits de destino ou transcodificado). O software FFMPEG disponibiliza ferramentas para isso.

Desta forma, foi feito um transcoder utilizando o FFMPEG onde duas sequências de vídeo foram definidas como entradas do sistema. As sequências são arquivos YUV de 1920 por 1080 pixels de resolução a uma taxa de quadros por segundo de 60 fps (para a primeira sequência) e 50 fps (para a segunda sequência). Foram levadas em consideração as degradações temporal e espacial no vídeo de entrada, bem como a transcodificação para H.264, simulando os diversos vídeos recebidos diariamente em um ambiente de emissora de TV.

Com o intuito de analisar e maximizar a qualidade dos vídeos encaminhados por telespectadores no processo de conversão de codificação do arquivo, foram realizados testes de qualidade com três diferentes métricas (PSNR, SSIM e VQM) para que se pudesse avaliar a melhor alternativa a ser empregada, usando a ferramenta livre, *open-source* e gratuita FFMPEG e, desta forma, transmitir o vídeo recebido em programas televisionados de forma otimizada.

Para testar, foram criados quatro cenários de aplicação simulando situações onde há degradação espacial e temporal (cenários 1 e 2), além de cenários onde foi possível analisar os efeitos da transcodificação na avaliação de qualidade do vídeo recuperado (cenários 3 e 4). Para as degradações temporais, foram testados nos cenários 2 e 4 o uso de um filtro de vídeo que implementa um algoritmo de predição de movimento (próprio do FFMPEG), chamado *minterpolate*, fim de se comparar os resultados sem a presença do mesmo.

Finalmente, os resultados mostram que o FFMPEG possui um bom desempenho no que diz respeito à transcodificação e que o uso do algoritmo de predição de movimento otimiza consideravelmente os resultados e a visualização das mídias pelo telespectador. Além disso, conclui-se que o FFMPEG trata melhor as degradações temporais (com predição de movimento) especialmente para mídias onde há movimento de *pan*. Quando não se utiliza o filtro *minterpolate*, os casos de degradação temporal apresentam melhores resultados para mídias com movimento de *pan* e *tilt*.

ABSTRACT

In a television broadcast environment, hundreds of videos recorded by different devices are received daily in various formats, which in most cases are not in the desired pattern for the transmission to take place.

The technical solution that allows video transmission is called video transcoding. Transcoding is the process of converting the video format, converting a compressed bit stream (called source bitstream or input) to another compressed bit stream (called a destination or transcoded bit stream). The FFMPEG software provides tools for this.

In this way, a transcoder was made using the FFMPEG where two video sequences were defined as system inputs. The sequences are YUV files of 1920 by 1080 pixel resolution at a frame rate of 60 fps (for the first sequence) and 50 fps (for the second sequence). Temporal and spatial degradations were taken into account in the input video as well as transcoding to H.264, simulating the various videos received daily in a TV broadcast environment.

In order to analyze and maximize the quality of the videos sent by viewers in the file encoding conversion process, quality tests were performed with three different metrics (PSNR, SSIM and VQM) in order to evaluate the best alternative to be used, using the free tool, *open-source* and free FFMPEG and in this way transmit the received video in televised programs in an optimized way.

In order to test, four application scenarios were created simulating situations where there is spatial and temporal degradation (scenarios 1 and 2), as well as scenarios where it was possible to analyze the effects of transcoding on the quality evaluation of the recovered video (scenarios 3 and 4). For the temporal degradations, the use of a video filter which implements a motion prediction algorithm (of FFMPEG library), called *minterpolate*, was tested in scenarios 2 and 4, in order to compare the results without the presence of the same.

Finally, the results show that the FFMPEG has a good performance regarding the transcoding and that the use of the motion prediction algorithm considerably optimizes the results and visualization of the media by the viewer. In addition, it is concluded that the FFMPEG treats temporal degradations better (with motion prediction) especially for media where there is *pan* movement. When the *minterpolate* filter is not used, the cases of temporal degradation present better results for media with *pan* and *tilt* movement.

SUMÁRIO

1	Introdução	1
1.1	CONTEXTUALIZAÇÃO.....	1
1.2	OBJETIVOS	2
1.3	ORGANIZAÇÃO DO TRABALHO.....	2
2	Fundamentação Teórica	3
2.1	IMAGEM.....	3
2.2	VÍDEO.....	3
2.2.1	RESOLUÇÃO DE UM VÍDEO	4
2.3	CODIFICAÇÃO DE VÍDEO	5
2.3.1	TIPOS DE CODIFICADORES	6
2.4	TRANSCODIFICAÇÃO DE VÍDEO	7
2.5	O SOFTWARE FFMPEG	8
2.6	MÉTRICAS DE QUALIDADE	9
2.6.1	CLASSIFICAÇÃO DAS MÉTRICAS.....	10
2.6.2	PSNR.....	10
2.6.3	SSIM	11
2.6.4	VQM	12
2.7	O ARQUIVO YUV	13
2.8	O PADRÃO H.264	14
3	Cenário de aplicação	15
4	Metodologia	17
4.1	CENÁRIO 1	17
4.2	CENÁRIO 2	18
4.3	CENÁRIO 3	18
4.4	CENÁRIO 4	19
5	Resultados	21
5.1	SEQUÊNCIAS UTILIZADAS	21
5.2	ANÁLISE DOS RESULTADOS	21
5.2.1	CENÁRIO 1	22
5.2.2	CENÁRIO 2	23
5.2.3	CENÁRIO 3	24
5.2.4	CENÁRIO 4	26
5.3	COMPARATIVO ENTRE DEGRADAÇÃO ESPACIAL E TEMPORAL	27

6	Conclusão.....	29
	REFERÊNCIAS BIBLIOGRÁFICAS.....	30
	Apêndices.....	31

LISTA DE FIGURAS

2.1	Representação de vídeo.....	4
2.2	Diagrama codificador	6
2.3	Funções da transcodificação.(1)	8
2.4	Diagrama VQM	13
3.1	Cenário 1.....	15
4.1	Cenário 1.....	17
4.2	Cenário 2.....	18
4.3	Cenário 3.....	18
4.4	Cenário 4.....	19
5.1	Primeiros quadros das sequências.....	21
5.2	DuckTakesOff reconstruído 384x216	22
5.3	BQTerrace reconstruído 384x216	22
5.4	BQTerrace reconstruído 6fps sem predição.....	24
5.5	BQTerrace reconstruído 6fps com predição	24
5.6	Comparativo dos cenários 1 e 3 - BQTerrace	25
5.7	Comparativo dos cenários 1 e 3 - DucksTakeOff	25
5.8	Comparativo dos cenários 2 e 4 - BQTerrace	26
5.9	Comparativo dos cenários 2 e 4 - DucksTakeOff	27
5.10	Comparativo degradação espacial e temporal - BQTerrace	28
5.11	Comparativo degradação espacial e temporal - DucksTakeOff	28

LISTA DE TABELAS

5.1	Resultados obtidos para o cenário 1	22
5.2	Resultados obtidos para o cenário 2 replicando frames	23
5.3	Resultados obtidos para o cenário 2 com o algoritmo de predição	23
5.4	Resultados obtidos para o cenário 3	24
5.5	Resultados obtidos para o cenário 4 replicando frames	26
5.6	Resultados obtidos para o cenário 4 com o algoritmo de predição	26
1	Dados obtidos para a sequência BQTerrace.....	31
2	Dados obtidos para a sequência DuckTakesOff.....	31

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

O Brasil passa por um processo de transformação em seu sistema de televisão. O sinal analógico, que era transmitido pelo país desde 1950 dá lugar à TV digital, advento tecnológico que muda a maneira com que o telespectador se relaciona com a TV.

O Sistema Brasileiro de Televisão Digital, SBTVD foi criado em 2006 e foi baseado no padrão japonês ISDB-T, o qual utiliza métodos avançados de codificação de áudio e vídeo, permitindo a recepção do sinal em dispositivos móveis e garantindo melhor qualidade de áudio e imagem quando comparado a outros sistemas. O SBTVD foi o primeiro sistema de TV aberta a utilizar o H264 com alta qualidade.

Paralelamente ao avanço da TV digital, tem-se notado um aumento radical da acessibilidade a celulares e câmeras de qualidade por toda a população brasileira, havendo um aumento na quantidade de mídias disponíveis. Desta forma, tem-se infinitos repórteres e cinegrafistas amadores ao redor do mundo e em tempo integral.

Para fazer uso das mídias advindas de diversos usuários e dispositivos, torna-se necessário garantir que o sistema implementado nas emissoras suporte-as em seus variados formatos e garantir uma certa qualidade nas mesmas para serem entregues aos telespectadores em diversas telas.

A solução técnica que permite a transmissão de vídeo se chama transcodificação de vídeo. A transcodificação é o processo que consiste na conversão do formato do vídeo, converte um fluxo de bits compactado (chamado fluxo de bits de origem ou entrada) para outro fluxo de bits compactado (chamado de fluxo de bits de destino ou transcodificado). Há diversos softwares que executam a transcodificação, um deles é o FFMPEG, framework comandado por linha de comando que suporta os mais variados tipos de multimídias existentes atualmente.

A avaliação da qualidade de um vídeo pode ser feita através da análise subjetiva ou de uma forma objetiva, utilizando métricas de avaliação de qualidade. A medida subjetiva da qualidade de um vídeo pode ser altamente desafiadora pois vários fatores e parâmetros das condições de exibição do vídeo, tais como iluminação da sala, brilho da tela, contraste, resolução e distância da tela, por exemplo, podem influenciar no resultado. Quanto à avaliação objetiva da qualidade de um vídeo, essa tem por objetivo o desenvolvimento de medidas quantitativas que possam definir automaticamente a qualidade percebida da imagem. Trata-se de modelos matemáticos ou equações cujos resultados devem correlacionar-se bem com avaliações subjetivas.

1.2 OBJETIVOS

O objetivo deste trabalho é analisar as características e o poder de transcodificação do software FFMPEG na recuperação de uma sequência de vídeo através do uso de três métricas objetivas de avaliação de qualidade: PSNR, SSIM e VQM (as quais serão detalhadas no capítulo 2).

1.3 ORGANIZAÇÃO DO TRABALHO

Este trabalho está sumarizado nos seguintes capítulos:

- Capítulo 2 São apresentados conceitos importantes para o melhor entendimento da análise realizada, bem como a melhor compreensão dos resultados obtidos.
- Capítulo 3 - Descreve o cenário de aplicação onde a análise foi realizada.
- Capítulo 4 - Descreve a metodologia utilizada e os procedimentos que foram realizados para o levantamento de dados.
- Capítulo 5 - Resultados, apresenta os resultados para os vídeos analisados de acordo com as métricas de avaliação de qualidade.
- Capítulo 6 - Conclusões e propostas para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

O objetivo deste capítulo é explicar os principais conceitos utilizados para que se possa compreender a análise realizada do software FFMPEG. Dentre os fundamentos os quais serão abordados estão os conceitos de Imagem, Vídeo, Codificação de Vídeo, Transcodificação de Vídeo e Métricas de Qualidade.

2.1 IMAGEM

A maioria das imagens é gerada por uma combinação de uma fonte de iluminação e da reflexão, absorção, refração ou difração da energia dessa fonte por elementos da cena (2).

Uma cena pode ser definida como uma entidade tridimensional formada a partir da reflexão de energia radiante. Um fluxo de radiação de uma fonte, ao se propagar pelo espaço, pode interagir com a superfície dos objetos de uma cena. A energia absorvida por tais objetos geralmente causa aquecimento. Já a energia refletida carrega informações a respeito desses objetos. Podem-se registrar tais informações com sensores que sejam sensíveis à faixa de radiação refletida (2). Embora uma cena real seja uma entidade tridimensional, uma imagem pode ser representada por uma função bidimensional, $f(x, y)$, em que x e y são coordenadas espaciais.

Uma imagem pode ser contínua com respeito às coordenadas espaciais x e y , e com respeito à intensidade $f(x, y)$. Assim, para criar uma imagem digital, é necessário converter esses dados contínuos para a forma digital. O processo de digitalização é resultado da amostragem, que é a discretização das coordenadas espaciais, e da quantização, que é a discretização da intensidade. A amostragem consiste em discretizar o domínio de definição da função $f(x, y)$, transformando-o em uma grade de pontos regularmente espaçados entre si. A quantização, por sua vez, define um conjunto finito de níveis de cinza ou níveis de cores com os quais uma imagem pode ser representada (2).

2.2 VÍDEO

Um sinal de vídeo é qualquer sequência de imagens que variam com o tempo. Uma imagem estática é uma distribuição espacial de intensidades que permanecem constantes com o tempo, enquanto uma imagem que varia com o tempo possui uma distribuição espacial de intensidades que varia com o tempo. Um sinal de vídeo é tratado como uma série de imagens chamados frames ou quadros (3). A taxa de frames (do termo em inglês frame rate) é a frequência com que os frames são exibidos e criam a ilusão de um vídeo

contínuo.

Em um vídeo digital, a informação passa por uma amostragem temporal, uma amostragem espacial e as intensidades resultantes dos pixels são quantizadas. Assim, a amostragem espacial e a quantização são as mesmas descritas para as imagens na seção anterior. Porém, a amostragem do vídeo envolve tirar amostras de uma nova dimensão: o tempo. A dimensão do tempo tem uma direção associada, ao contrário das dimensões espaciais, cujo sistema de coordenadas é artificialmente imposto. O tempo procede do passado em direção ao futuro, com uma origem que existe apenas no momento corrente (4). O resultado das amostras no tempo são os frames citados anteriormente, ou uma série de imagens completas, cada qual composto de amostras espaciais.

Dessa forma, cada frame de um vídeo pode ser representado por uma função $f(x, y, t)$ ou $I(x, y, t)$, como representado na Figura 2.1. Por exemplo, para o instante $t = t_0$, tem-se:

$$f(x, y, t_0) = \begin{bmatrix} f(0, 0, t_0) & f(0, 1, t_0) & \dots & f(0, N-1, t_0) \\ f(1, 0, t_0) & f(1, 1, t_0) & \dots & f(1, N-1, t_0) \\ \vdots & \vdots & \ddots & \vdots \\ f(M-1, 0, t_0) & f(M-1, 1, t_0) & \dots & f(M-1, N-1, t_0) \end{bmatrix}$$

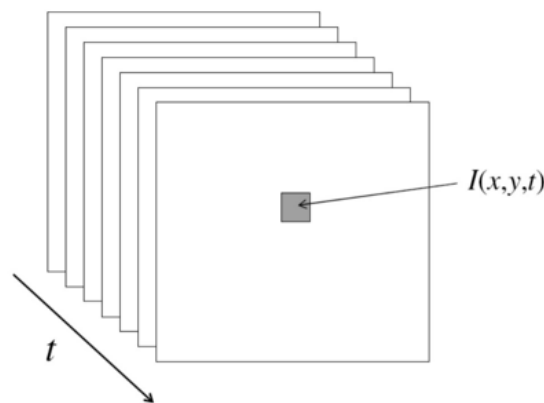


Figura 2.1: Representação de um vídeo. (5)

2.2.1 Resolução de um vídeo

Um conceito importante para o entendimento é resolução do vídeo. Resolução de vídeo em um dispositivo de exibição refere-se ao número de pixels distintos que podem ser exibidos em cada dimensão. Geralmente é citado como largura x altura; para um exemplo: "1024 x 768". Neste exemplo, 1024 indica a largura e 768 indica a altura em que a exibição pode ser resolvida em pixels. Uma resolução de tela é controlada por fatores como o tubo de raios catódicos e o monitor de tela plana que são apresentados em televisões digitais e monitores de computador.

A resolução espacial refere-se ao número de pixels utilizados na construção da imagem. Imagens com maior resolução espacial são compostas por um maior número de pixels do que

aquelas de menor resolução espacial. Descreve quantos detalhes em uma imagem fotográfica são visíveis ao olho humano. Quanto à resolução temporal de um vídeo, é um conceito relacionado ao nível de detalhe por unidade de tempo.

Para o melhor entendimento dos conceitos descritos acima, imagine um fluxo de dados de vídeo em 1920 x 1080 pixels, 60 quadros por segundo (um formato HDTV comum). Em teoria, isso significa que a imagem original foi amostrada espacialmente como 1920 x 1080 pixels (um número igual de amostras por unidade de distância em cada direção, assumindo uma proporção de 16: 9), e cada ponto de amostragem (pixel) também está sendo amostrado 60 vezes por segundo. A resolução espacial determina quantos detalhes podemos resolver em cada imagem estática individual ou “quadro” tirado daquele conjunto, e a resolução temporal (a taxa de amostragem no tempo, que é 60 amostras por segundo) determina a mudança mais rápida que podemos rastrear entre imagens sucessivas

2.3 CODIFICAÇÃO DE VÍDEO

A codificação de vídeo é o processo em que uma compressão do sinal de vídeo é realizada para que o mesmo possa ser transmitido de acordo com o canal desejado, de acordo com a largura de banda do mesmo. Desta forma, o processo de codificação tem fundamental importância para a transmissão de mídias bem como no que diz respeito ao armazenamento das mesmas.

A configuração de um sistema completo de codificação de vídeo pode ser organizada de acordo com o diagrama de blocos ilustrado na Fig. 2.2. Os blocos de construção do sistema de codificação de vídeo são resumidos abaixo:

- Aquisição de Vídeo: Origem da sequência de vídeo, a qual é produzida em um formato digital.
- Pré-processamento: Operações no material de origem de vídeo bruto não comprimido, como redução de número de frames, conversão de formato de cor, correção de cor ou eliminação de ruído.
- Codificação - Transformação da sequência de entrada de vídeo em um fluxo de bits codificado. O objetivo da codificação é gerar uma representação compacta do vídeo de entrada que é adequada para o método de transmissão na aplicação dado o cenário.
- Transmissão: Empacotamento do fluxo de bits em um formato para transmissão da informação pelo canal. Inclui envio e entrega dos dados do vídeo para o lado do receptor, bem como métodos para proteção e recuperação de perdas.
- Decodificação: Transformação do fluxo de bits recebido em uma sequência de vídeo reconstruída. Como a codificação de vídeo geralmente requer compactação com

perdas para atingir as restrições de taxa de bits de transmissão de destino, o vídeo decodificado é uma aproximação do vídeo de origem original. Se ocorreram perdas de transmissão irreversíveis, o decodificador aplica estratégias de ocultação para recuperar a sequência de vídeo corrompida, tanto quanto possível.

- Pós-processamento: Operações na sequência de vídeo reconstruída para melhoramento ou para adaptação da sequência para exibição. Estas operações incluem correção de cor, corte ou reamostragem. Também efeitos especiais podem ser aplicados conforme determinado pela aplicação.
- Exibição-Apresentação da sequência de vídeos para visualização: A sequência de vídeo precisa ser transferida para o formato de cor apropriado para exibição. O tempo de saída das imagens da sequência de vídeo é um aspecto importante para alcançar a impressão visual pretendida (6).

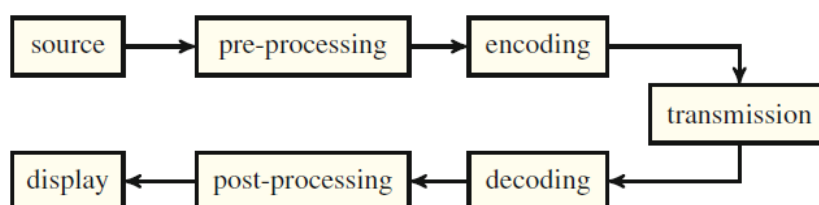


Figura 2.2: Diagrama de blocos de um codificador.(6)

2.3.1 Tipos de codificadores

Os codificadores podem ser divididos em dois grandes grupos: sem perdas (lossless) e com perdas (lossy). Os codificadores do primeiro grupo diminuem o tamanho de um arquivo sem alterar a qualidade do som e da imagem. Esse processo é chamado de compressão. Quando feita a descompressão, processo contrário a compressão, será obtido um arquivo exatamente igual ao original (geralmente, os arquivos comprimidos são de duas a três vezes menores que o original). O segundo grupo é formado por codecs que podem diminuir ainda mais o tamanho dos arquivos originais, chegando a gerar arquivos até 300 vezes menores, no caso de vídeos e 13 vezes menores no caso de arquivos de áudio. Neste caso, o arquivo codificado apresenta uma perda na qualidade do som e da imagem e não é possível obter exatamente o mesmo arquivo quando feita a decodificação.

A taxa de compressão é escolhida a fim de controlar o tamanho do arquivo final de acordo com a qualidade que se deseja. A transmissão esportiva como a de uma partida de futebol, por exemplo, deve codificar bem os movimentos, mas não precisa se preocupar muito com as cores ou texturas das imagens, já um leilão de joias precisaria se preocupar muito mais com estes últimos aspectos do que com a estimação ou predição de movimentos.

Aplicações como televisão digital ou os discos de alta definição vem substituindo os atuais

discos de DVD têm seus arquivos codificados bem menores que os originais apresentando uma perda imperceptível aos olhos humanos.

2.4 TRANSCODIFICAÇÃO DE VÍDEO

Por definição, a transcodificação é o processo que converte de um fluxo de bits compactado (chamado fluxo de bits de origem ou entrada) para outro fluxo de bits compactado (chamado de fluxo de bits de destino ou transcodificado). Várias propriedades podem mudar durante a transcodificação: o formato de vídeo, a taxa de bits do vídeo, a taxa de quadros, a resolução espacial, as ferramentas de codificação usadas, isto é, um bitstream pode usar certos quadros, enquanto o outro não, ou camadas de escalabilidade são adicionadas ao fluxo de bits alvo, e até mesmo a inserção de novas informações no vídeo, como marcas d'água, dados ocultos ou uma camada para resiliência de erro (7).

No cenário de aplicação da transcodificação, uma ou mais características do fluxo de bits de vídeo precisam ser alteradas para permitir a comunicação entre os dois sistemas. Para permitir a interoperação de conteúdo multimídia, a transcodificação é necessária tanto dentro quanto através de diferentes formatos (7).

Existem três premissas básicas na transcodificação:

- A informação do bitstream original deverá ser explorado o máximo possível;
- A qualidade do vídeo resultante do novo bitstream deverá ser a mais alta possível;
- Em aplicações em tempo real, o atraso da transcodificação e requisitos de memória deverão ser minimizados para que o processo seja fiel à aplicação.

Um transcodificador de vídeo pode fornecer funções como o ajuste da taxa de bit e conversão de formato. Um resumo destas funções citadas acima está ilustrada na figura 2.3.

A transcodificação homogênea executa a conversão entre bitstreams de vídeo do mesmo padrão. Um vídeo de origem de alta qualidade pode ser transcodificado para um bitstream de vídeo de baixa qualidade, com diferentes resoluções espaciais / temporais e diferentes taxas de bits. Uma técnica simples para transcodificar um vídeo para diminuir a taxa de bits é aumentar o passo de quantização na parte do codificador no transcodificador.

Quanto ao transcodificador de vídeo heterogêneo, o mesmo fornece conversões entre padrões de codificação de vídeo existentes e futuros. Ele fornece conversão de sintaxe entre esses padrões. Além disso, um transcodificador de vídeo heterogêneo também pode fornecer as funcionalidades da transcodificação homogênea. A transcodificação pode incluir funções adicionais, como a resiliência de erro e a inserção de logotipo ou marca d'água.

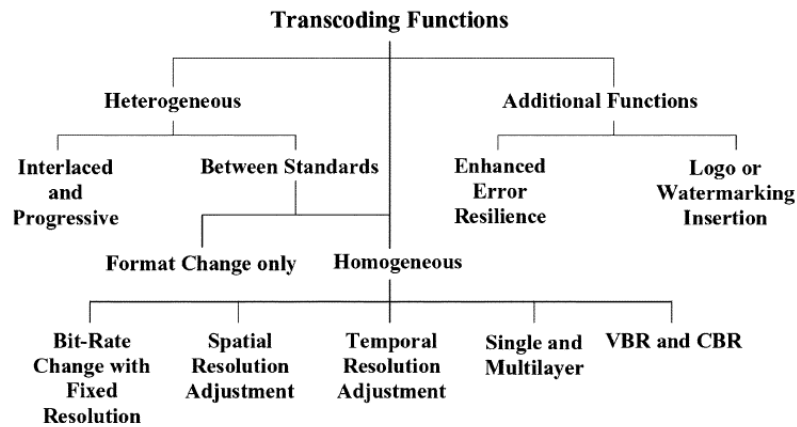


Figura 2.3: Funções da transcodificação.(1)

Uma das primeiras aplicações da transcodificação é adaptar a taxa de bits de um fluxo de vídeo pré-comprimido para uma largura de banda de canal. Por exemplo, um programa de TV pode ser originalmente compactado em uma alta taxa de bits para aplicativos de estúdio, mas depois precisa ser transmitido por um canal com uma taxa de bits muito menor. No acesso universal às mídias, diferentes dispositivos podem ter diferentes acessos à Internet, incluindo rede de acesso local (LAN), linha de assinante digital (DSL), cabo, redes sem fio, rede digital de serviços integrados (ISDN). As diferentes redes de acesso têm diferentes características de canal, como largura de banda, taxas de erro de bit e taxas de perda de pacote. Os dispositivos utilizados pelo usuário final (notebooks, smartphones, tablets) estão substituindo os computadores pessoais como os terminais dominantes de acesso à Internet. Esses dispositivos variam significativamente em recursos, como capacidade de computação e capacidade de exibição. Para entregar de forma flexível dados multimídia a usuários com diferentes recursos disponíveis, redes de acesso e interesses, o conteúdo multimídia pode precisar ser adaptado dinamicamente de acordo com o ambiente de uso. A transcodificação é uma das principais tecnologias para realizar essa tarefa desafiadora e também é útil para a adaptação de conteúdo para redes peer-to-peer em links de comunicação compartilhados multihop (8).

2.5 O SOFTWARE FFMPEG

O Fast Forward MPEG (FFmpeg) é um framework, comandado por linha de comando, que suporta os mais variados tipos de multimídias existentes atualmente. É utilizado para codificação, decodificação, conversão, análise e exibição destes formatos, além de suportar aquisição e codificação destas multimídias em tempo real.

Já foi utilizado amplamente pela comunidade acadêmica para desenvolver diferentes tipos de soluções, como: codec de áudio e vídeo para a plataforma Android, streaming HLS (HTTP Live Streaming) e módulo para inspeção automática de linhas de transmissão

por termografia. Ou seja, é um software amplamente testado, tornando-se, de certa forma, confiável (9).

O software se trata de um framework de multimídia licenciado sob a GNU Lesser General Public License (LGPL), ou seja, é licenciado para ser utilizado em programas próprios, porém, as otimizações opcionais só permitem a utilização em programas livres.

Segundo sua descrição, o FFmpeg é capaz de trabalhar com a maioria das multimídias criadas até o momento. Ferramentas disponíveis:

- Encode: Transforma multimídias de formatos primitivos em multimídias de formatos comprimidos. Estes formatos comprimidos são aqueles onde as redundâncias da multimídia primitiva são modificadas para transmissão ou armazenamento;
- Decode: Processo inverso ao encode, porém a multimídia resultante contém as modificações efetuadas pelo encode, diferenciando o resultado final em comparação ao formato primitivo da multimídia;
- Transcode: Transforma multimídias de formatos comprimidos em outro formato comprimido;
- Mux: De modo simples refere-se à aglutinação de streams em um único arquivo, como, por exemplo, a união de um stream de vídeo e um stream de áudio;
- Demux: Processo inverso do Mux, de modo simples, separa os streams contidos em um arquivo;

2.6 MÉTRICAS DE QUALIDADE

A qualidade de um vídeo pode ser medida de acordo com o relato da experiência dos espectadores, o que usualmente pode ser mensurado com confiança por métodos de análise subjetivos. Vários estudos vem sendo realizados para desenvolver métodos objetivos para avaliar a qualidade de um vídeo onde haja alta correlação com os resultados de avaliação subjetiva.

A medida subjetiva da qualidade de um vídeo pode ser altamente desafiadora pois vários fatores e parâmetros das condições de exibição do vídeo, tais como iluminação da sala, brilho da tela, contraste, resolução e distância da tela, por exemplo, podem influenciar no resultado. Através da recomendação ITU-T BT.500-11 (2002) “Metodologia para a avaliação subjetiva da qualidade de imagens de televisão”, alguns métodos de teste, escalas de classificação e condições de visualização foram padronizados para que resultados significativos fossem obtidos na análise subjetiva. Estes métodos de teste são classificados em métodos de duplo estímulo e de estímulo único, onde nos de duplo estímulo os espectadores

avaliam a qualidade ou a alteração de qualidade entre dois fluxos de vídeo (o de referência e o alterado) e nos de estímulo único avaliam apenas o fluxo de vídeo alterado.

Quanto à avaliação objetiva da qualidade de um vídeo, essa tem por objetivo o desenvolvimento de medidas quantitativas que possam definir automaticamente a qualidade percebida da imagem. Trata-se de modelos matemáticos ou equações cujos resultados devem correlacionar-se bem com avaliações subjetivas. A qualidade de uma métrica objetiva de qualidade de vídeo pode ser avaliada através do cálculo da correlação entre as pontuações objetivas e os resultados subjetivos dos testes. Os coeficientes de correlação mais utilizados são o coeficiente de correlação linear de Pearson e o coeficiente de correlação de ordem de Spearman.

Métricas de qualidade de vídeo são medidas objetivas que tem por objetivo avaliar a qualidade de um vídeo de forma quantitativa.

2.6.1 Classificação das métricas

As métricas objetivas tem como base o vídeo original, de alta qualidade, o qual serve como referência, e são classificadas como: referência completa (FR), referência reduzida (RR), e sem referência (NR), onde:

- Referência completa (FR): calculam uma função da diferença entre cada pixel em cada quadro do vídeo de teste e seu pixel correspondente no vídeo de referência.
- Referência reduzida (RR): extraem alguns recursos de ambos os vídeos e os comparam para fornecer um índice de qualidade. Apenas alguns recursos do vídeo de referência devem ser enviados junto com o vídeo compactado para avaliar a qualidade de vídeo recebida no final do receptor.
- Sem referência (NR): avaliam a qualidade de um vídeo de teste sem qualquer referência ao vídeo original (3).

Neste trabalho, trataremos das métricas de qualidade objetivas PSNR, VQM e SSIM, as quais serão utilizadas juntamente ao software FFMPEG para que se possa avaliar, sob diferentes cenários, a capacidade de transcodificação do mesmo.

2.6.2 PSNR

Peak Signal to Noise Ratio ou a relação sinal/ruído de pico é uma métrica que tem como base o erro quadrático médio (10). A PSNR calcula o logaritmo da razão entre a potência máxima do sinal e a diferença quadrática média (MSE) entre um quadro de vídeo original e o quadro do vídeo prejudicado/comprimido. A suposição por trás do PSNR é que a diferença na qualidade perceptual entre uma imagem I e uma imagem de referência

R (geralmente a imagem original, não comprimida) está relacionada à força do sinal de erro entre essas duas imagens (7).

$$PSNR = 10 * \log_{10} \left(\frac{L^2}{MSE} \right) \quad (2.1)$$

A PSNR pode ser obtida através da fórmula da equação 2.1, onde L é o número de bits da imagem. A unidade de medida para a PSNR é o decibél e, quanto maior o valor, maior será a semelhança da imagem. Geralmente, em situações práticas, o valor da PSNR se limita a 50dB.

Tendo em vista a facilidade de cálculo, a PSNR é uma das métricas mais utilizadas (dado que se tenha o vídeo de referência) para se calcular a distorção entre imagens e vídeos. Entretanto, a mesma sofre de algumas limitações como, por exemplo, não se correlacionar bem com medidas subjetivas de qualidade de vídeo. Duas imagens distorcidas com o mesmo PSNR podem ter tipos muito diferentes de erros e, portanto, podem ter qualidades perceptivas muito diferentes. Para uma determinada imagem ou sequência de imagens, PSNR alto geralmente indica alta qualidade e PSNR baixo geralmente indica baixa qualidade. No entanto, um valor particular de PSNR não significa necessariamente uma qualidade subjetiva "absoluta"(10). Apesar disso, estudos indicam que a PSNR se correlaciona bem com medidas subjetivas em certos cenários.

2.6.3 SSIM

A SSIM é projetada para medir a mudança percebida no conteúdo (7) ao contrário da PSNR, que calcula a distorção real no conteúdo. SSIM pode ser traduzida como semelhança estrutural e a métrica pode ser calculada em várias janelas de uma imagem.

Dadas duas imagens x e y , a métrica SSIM é obtida através da equação 2.2:

$$SSIM(x, y) = l(x, y) * t(x, y) \quad (2.2)$$

O primeiro termo, $l(x, y)$, indica a similaridade da luminância. Este termo é calculado a partir da média harmônica de $x|y$ e $y|x$, como segue (11):

$$l(x, y) = \frac{(2\mu_x\mu_y + C_1)}{(\mu_x^2 + \mu_y^2 + C_1)} \quad (2.3)$$

Onde μ_x e μ_y são as médias dos dois blocos de imagem de entrada. C_1 é um fator de regularização para pequenos valores médios. O segundo termo, $t(x, y)$, indica a similaridade da textura, (equação 2.4).

$$t(x, y) = \frac{(2\sigma_{xy} + C_2)}{(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (2.4)$$

Onde σ_{xy} é a covariância entre x e y e C_2 também é um fator de regularização para pequenos valores médios. É proposto pelos autores (12) usar $C_1 = (0,01L)^2$ e $C_2 = (0,03L)^2$, onde L é o valor máximo possível para a imagem.

A SSIM é uma métrica simples, se comporta melhor que PSNR em alguns casos por avaliar as distorções estruturais da imagem. Essa métrica não compara valores de pixel, mas os elementos de imagem percebidos pelo olho humano. A SSIM descreve melhor as diferenças de qualidade de imagem do que a PSNR, conforme detalhado no capítulo 5.

Os valores da SSIM variam de -1 a 1. Quanto mais próximo de 1 for o valor encontrado, melhor qualidade de imagem (13).

2.6.4 VQM

O VQM - *Video Quality Model* - foi projetado para ser um modelo de qualidade de vídeo de propósito geral para sistemas de vídeo que abrangem uma ampla variedade de taxas de qualidade e bit. O Modelo Geral utiliza tecnologia de referência reduzida e fornece estimativas das impressões gerais de qualidade de vídeo (ou seja, pontuações de opinião médias, produzidas por painéis de espectadores). Suas técnicas de calibração associadas compreendem um sistema de medição de qualidade de vídeo objetivo e automatizado completo. A calibração dos fluxos de vídeo originais e processados inclui alinhamento espacial, estimativa de região válida, cálculo de compensação de ganho e nível e alinhamento temporal (14). O diagrama de blocos do VQM está ilustrado na figura 2.4 abaixo.

O modelo geral contém sete parâmetros:

- ***si loss***: detecta a perda ou a diminuição de informação espacial.
- ***hv loss***: detecta um deslocamento das bordas da orientação horizontal e vertical para a orientação diagonal, como pode ser o caso se as bordas horizontais e verticais sofrerem mais desfocagem do que as bordas diagonais.
- ***hv gain***: esse parâmetro detecta uma mudança de arestas de diagonal para horizontal e vertical.
- ***chroma spread***: detecta mudanças na distribuição da distribuição de amostras de cores bidimensionais.
- ***si gain***: é o único parâmetro de melhoria de qualidade no modelo, mede melhorias na qualidade que resultam de aprimoramentos ou aprimoramentos de borda.
- ***ct ati gain***: identifica as deficiências na borda em movimento que estão quase sempre presentes, como o ruído da borda.

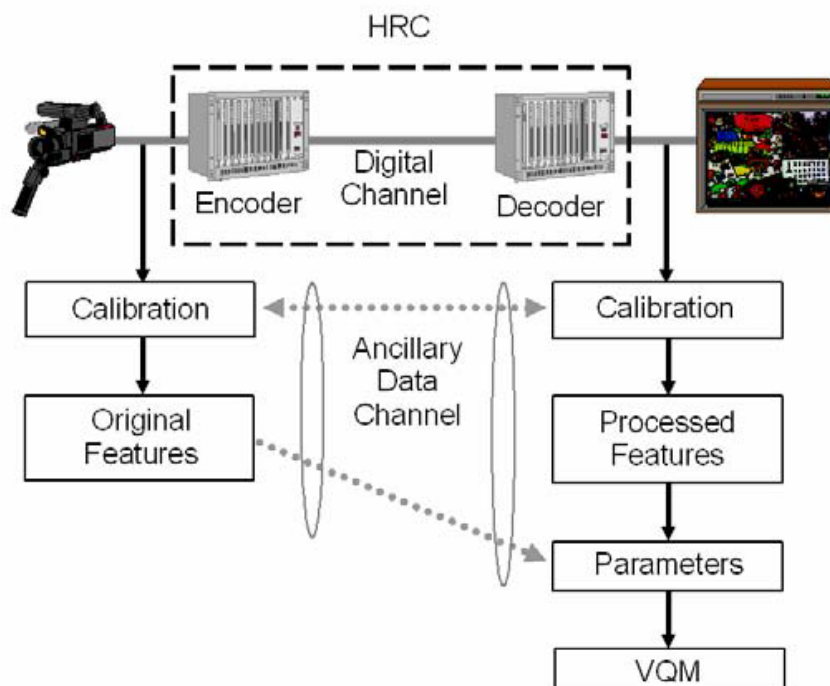


Figura 2.4: Diagrama de blocos do VQM. (14)

- **chroma extreme**: usa os mesmos recursos de cor da métrica chroma spread, mas diferentes funções de recolhimento espaço-temporal. Detecta graves deficiências de cores localizadas, como aquelas produzidas por erros de transmissão digital.

Os valores de VQM são correlacionados com as classificações subjetivas do visualizador. Valores da métrica VQM variam entre 0 e 1. Valor 0 significa que não há diferença (melhor qualidade), e quanto mais próximo de 1, maior a diferença (pior qualidade) (13).

2.7 O ARQUIVO YUV

YUV é um sistema de codificação de cores. Ele codifica uma imagem colorida ou vídeo levando em consideração a percepção humana, permitindo largura de banda reduzida para componentes de cromaticidade e permitindo tipicamente que os erros de transmissão ou artefatos de compressão sejam mais eficientemente mascarados pela percepção humana. Outras codificações de cores têm propriedades semelhantes, e a principal razão para implementar ou investigar as propriedades do YUV seria a interface com a televisão analógica ou digital ou o equipamento fotográfico que esteja em conformidade com certos padrões YUV.

O parâmetro Y representa a luminância (informação em preto e branco), enquanto os parâmetros U e V indicam a cromaticidade, ou seja, o valor das cores. Este modelo foi desenvolvido para permitir a transmissão das informações coloridas para as televisões a

cores, garantindo que as existentes em preto e branco pudessem continuar a exibir a imagem em tons de cinza.

2.8 O PADRÃO H.264

O H.264 é um padrão da indústria para codificação de vídeo, bem como um formato popular para vídeo codificado, um conjunto de ferramentas para compactação de vídeo e um estágio em um ambiente de comunicação de vídeo digital em constante evolução.

É um método e formato para compactação de vídeo, o processo de conversão de vídeo digital em um formato que ocupa menos espaço quando é armazenado ou transmitido. A compactação de vídeo ou codificação de vídeo é uma tecnologia essencial para aplicativos como televisão digital, DVD-Video, TV móvel, videoconferência e transmissão de vídeo pela Internet. A padronização da compactação de vídeo possibilita que produtos de diferentes fabricantes, como codificadores, decodificadores e mídia de armazenamento, interajam.

O padrão H.264 / AVC foi publicado pela primeira vez em 2003, com várias revisões e atualizações publicadas desde então. Baseia-se nos conceitos de padrões anteriores, como MPEG-2 e MPEG-4 Visual, e oferece o potencial de melhor eficiência de compactação, ou seja, vídeo compactado de melhor qualidade e maior flexibilidade na compactação, transmissão e armazenamento de vídeo. Este padrão descreve um conjunto de ferramentas ou métodos para compactação de vídeo. Nele, é especificado como o vídeo codificado com essas ferramentas deve ser representado e decodificado. Um codificador de vídeo pode escolher quais ferramentas usar e como aplicá-las à sequência de vídeo atual, com algumas restrições.

Um dos motivos mais relevantes para a padronização do H.264 e sua adoção subsequente pela indústria é seu melhor desempenho em comparação com os padrões anteriores. O H.264 oferece desempenho de compactação significativamente melhor que o MPEG-2 Visual pois, ao se usar o H.264, é possível compactar o vídeo em um número muito menor de bits do que usando o MPEG-2, para a mesma resolução de vídeo e qualidade de imagem. Isso significa, por exemplo, que muito mais material de vídeo pode ser armazenado em um disco ou transmitido por um canal de transmissão usando o formato H.264.

Apesar de atualmente existirem padrões mais avançados do que o H264 em termos de eficiência de compressão (como o H265/HEVC por exemplo), o H264 ainda é um padrão bastante popular. Neste trabalho, ele é especialmente importante e será adotado como padrão devido ao fato de ser o padrão adotado no SBTVD. Desta forma, serão convertidos vídeos originalmente em YUV para H264 a fim de se testar a capacidade do software FFMPEG.

3 CENÁRIO DE APLICAÇÃO

Um conceito difundido nas emissoras de televisão é o de que, com o avanço e a acessibilidade de celulares e câmeras de boa qualidade e a difusão das redes sociais, tem-se infinitos repórteres e cinegrafistas gratuitos ao redor do mundo e em tempo integral, de tal forma que a quantidade de mídias disponíveis cresceu radicalmente, sendo também difícil não ter conhecimento de um evento.

Contudo, para fazer uso dessas mídias, torna-se necessário garantir que o sistema de Media Asset Management (MAM) implementado suporte-as em seus variados formatos e garantir uma certa qualidade nas mesmas para serem entregues aos telespectadores em diversas telas.

Há algumas soluções proprietárias de broadcast que são encarregadas da entrega de mídias para o formato padrão utilizado no MAM da emissora e algumas usam o transcodificador livre, gratuito e de código aberto FFMPEG.

O cenário de aplicação tem como base uma emissora de TV onde são encaminhados vários vídeos gravados pelos telespectadores em dispositivos e formatos diversos, vídeos que serão usados nos programas para aumentar a interação entre o público. Para que a transmissão dos vídeos recebidos possa ser realizada, uma série de características no vídeo precisa ser alterada para garantir também a qualidade e a boa visualização das mídias pelos telespectadores.

Para se ter uma ideia da diversidade de características dos vídeos recebidos diariamente em uma emissora de TV, a figura 3.1 ilustra que há uma grande variedade de tamanhos, resoluções e formatos.

Format	File Size	Bit Rate	Width	Height	Frame Rate Mode	Frame Rate	Min Frame Rate	Max Frame Rate
MPEG-4	8.20 MiB	1 440 kb/s	848	480	Variable	29,97	15	30
MPEG-4	13.1 MiB	1 039 kb/s	640	352	Constant	30	-	-
MPEG-4	644 KiB	727 kb/s	800	450	Variable	14,846	7,972	15,023
AVI	348 KiB	285 kb/s	352	240	Constant	15	-	-
MPEG-4	7.45 MiB	885 kb/s	1280	720	Constant	25	-	-
MPEG-4	3.67 GiB	V 59.9 Mb/s	1920	1080	Constant	59,94	-	-

Figura 3.1: Mídias recebidas em uma emissora de TV.

Este trabalho analisará tratamentos de mídias feito pelo FFMPEG sob o olhar de métricas de qualidade temporais, espaciais e de experiência, a fim de verificar a eficácia do transcoder no ambiente de produção de uma emissora de TV, averiguação que representa a possibilidade de uma grande redução de custos e o desenvolvimento de novas tecnologias.

Serão utilizados vídeos com características similares aos previamente ilustrados na figura

3.1 na entrada do transcoder e, com base na transcodificação feita pelo FFMPEG, o vídeo resultante da operação de transcodificação será avaliado (tendo como referência o vídeo original em YUV) através das métricas PSNR, SSIM e VQM.

Definimos como formato alvo o arquivo de saída do transcoder o arquivo H.264 com as seguintes características, tendo como base o padrão adotado em uma emissora de TV:

- Resolução espacial: 1920 x 1080p
- Resolução temporal: 30 quadros por segundo
- Taxa de bit: 50 Mbps

4 METODOLOGIA

Este capítulo apresenta o detalhamento dos métodos utilizados para avaliar a qualidade de transcodificação do software FFMPEG. Foram considerados quatro diferentes cenários para realizar a análise. As implementações foram feitas utilizando os softwares MatLab (versão R2017b), FFMPEG (versão 4.0) e BVQM (versão 2.1).

O objetivo principal dos métodos utilizados foi o de avaliar o poder de transcodificação do FFMPEG sob cenários onde houvesse degradação temporal e espacial de um arquivo YUV, bem como cenários onde, além das degradações previamente citadas, também houvesse a transcodificação do YUV degradado para o formato H.264 (com o intuito de se avaliar o impacto da alteração de formato).

As degradações espaciais e temporais foram feitas utilizando o software Matlab. Ao se importar o arquivo YUV são criadas 3 matrizes com onde a primeira representa a altura, a segunda a largura e a terceira o tempo de duração do vídeo. A degradação foi feita retirando informações dessas matrizes, ou seja, o vídeo foi deteriorado propositalmente.

A criação dos quarto cenários os quais serão explicados a seguir tem como objetivo a simulação, em ambiente controlado, das situações que ocorrem no mundo real.

4.1 CENÁRIO 1

Neste cenário faremos a análise do FFMPEG sob o caso da degradação espacial em um arquivo YUV, sem a transcodificação.

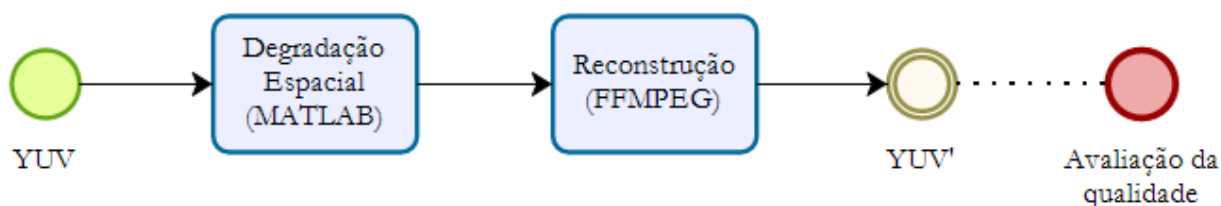


Figura 4.1: Diagrama do cenário 1.

- **Degradação espacial:** Alteração da resolução do arquivo YUV de entrada para $\frac{1}{2}$, $\frac{1}{3}$, $\frac{1}{4}$ e $\frac{1}{5}$ da resolução original.
- **Reconstrução:** Utilizando o FFMPEG, o vídeo deteriorado foi reconstruído. Nessa etapa, o software realizou o *upscaling* para recuperar vídeo deteriorado à qualidade original;
- **Arquivo YUV':** Arquivo resultante da reconstrução feita pelo FFMPEG;

- **Avaliação de qualidade:** Métricas de qualidade foram executadas para avaliar a qualidade da reconstrução do FFMPEG, onde é feita a comparação quadro a quadro entre o vídeo original (YUV) e o vídeo reconstruído (YUV')

4.2 CENÁRIO 2

Neste cenário faremos a análise do FFMPEG sob o caso da degradação temporal em um arquivo YUV, sem a conversão de formato do mesmo.

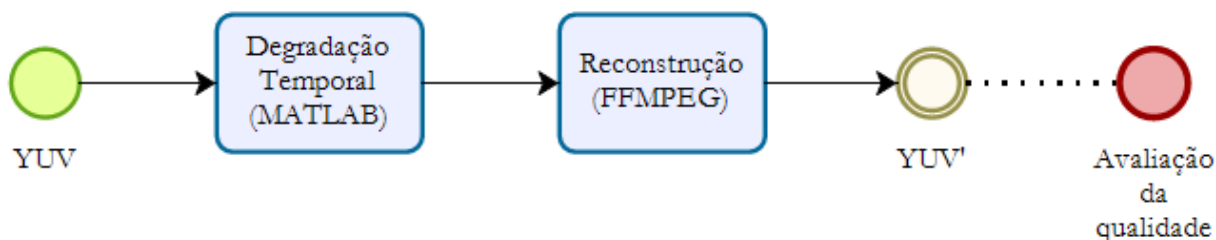


Figura 4.2: Diagrama do cenário 2.

- **Degradação temporal:** Alteração da taxa de quadros por segundo (fps) do arquivo YUV de entrada para $\frac{1}{2}$, $\frac{1}{3}$, $\frac{1}{5}$ e $\frac{1}{10}$ do fps original.
- **Reconstrução:** Utilizando o FFMPEG, o vídeo deteriorado foi reconstruído. Nessa etapa, o software realizou o *upsampling* para recuperar vídeo deteriorado à qualidade original;
- **Arquivo YUV':** Arquivo resultante da reconstrução feita pelo FFMPEG;
- **Avaliação de qualidade:** Métricas de qualidade foram executadas para avaliar a qualidade da reconstrução do FFMPEG, onde é feita a comparação quadro a quadro entre o vídeo original (YUV) e o vídeo reconstruído (YUV')

4.3 CENÁRIO 3

Neste cenário faremos a análise do FFMPEG sob o caso da degradação espacial em um arquivo YUV, com a transcodificação para o formato H.264 antes da análise de qualidade.

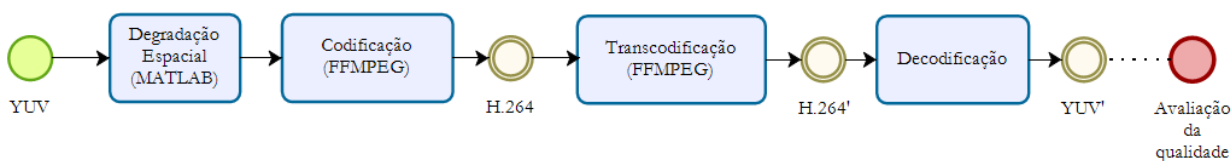


Figura 4.3: Diagrama do cenário 3.

- **Degradação espacial:** Alteração da resolução do arquivo YUV de entrada para $\frac{1}{2}$, $\frac{1}{3}$, $\frac{1}{4}$ e $\frac{1}{5}$ da resolução original.
- **Codificação:** Utilizando o FFMPEG, o vídeo deteriorado foi codificado para o formato H.264.
- **Arquivo H.264:** Arquivo resultante da codificação no FFMPEG; é o arquivo análogo ao recebido pela emissora de TV.
- **Transcodificação:** Utilizando o FFMPEG, o vídeo em H.264 foi transcodificado, resultando no arquivo H.264'. Nessa etapa, certas propriedades do vídeo foram alteradas (como resolução e fps) com o intuito de adequar o vídeo para a transmissão no sistema SBTVD.
- **Arquivo H.264':** Arquivo resultante da transcodificação.
- **Decodificação:** O fluxo de bits resultado da transcodificação foi decodificado para que o formato YUV novamente, para que o mesmo pudesse ser avaliado através das métricas, produzindo o arquivo YUV'. Ocorreu a transformação de um fluxo de bits para pixels.
- **Arquivo YUV':** Arquivo resultante da decodificação o qual será avaliado;
- **Avaliação de qualidade:** Métricas de qualidade foram executadas para avaliar a qualidade da transcodificação do FFMPEG, onde é feita a comparação quadro a quadro entre o vídeo original (YUV) e o vídeo reconstruído (YUV')

4.4 CENÁRIO 4

Neste cenário faremos a análise do FFMPEG sob o caso da degradação temporal em um arquivo YUV, com a transcodificação para o formato H.264 antes da análise de qualidade.

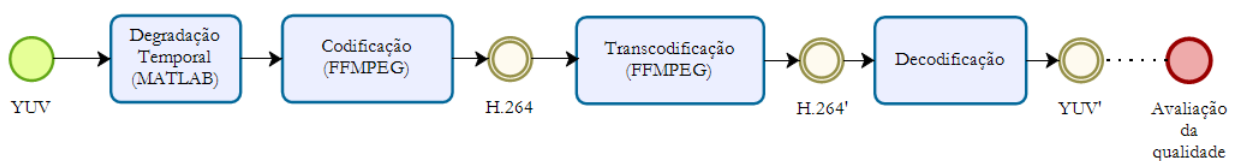


Figura 4.4: Diagrama do cenário 4.

- **Degradação temporal:** Alteração da taxa de quadros por segundo (fps) do arquivo YUV de entrada para $\frac{1}{2}$, $\frac{1}{3}$, $\frac{1}{5}$ e $\frac{1}{10}$ do fps original.
- **Codificação:** Utilizando o FFMPEG, o vídeo deteriorado foi codificado para o formato H.264.

- **Arquivo H.264:** Arquivo resultante da codificação no FFMPEG; é o arquivo análogo ao recebido pela emissora de TV.
- **Transcodificação:** Utilizando o FFMPEG, o vídeo em H.264 foi transcodificado, resultando no arquivo H.264'. Nessa etapa, certas propriedades do vídeo foram alteradas (como resolução e fps) com o intuito de adequar o vídeo para a transmissão no sistema SBTVD.
- **Arquivo H.264':** Arquivo resultante da transcodificação.
- **Decodificação:** O fluxo de bits resultado da transcodificação foi decodificado para que o formato YUV novamente, para que o mesmo pudesse ser avaliado através das métricas, produzindo o arquivo YUV'. Ocorreu a transformação de um fluxo de bits para pixels.
- **Arquivo YUV':** Arquivo resultante da decodificação o qual será avaliado;
- **Avaliação de qualidade:** Métricas de qualidade foram executadas para avaliar a qualidade da transcodificação do FFMPEG, onde é feita a comparação quadro a quadro entre o vídeo original (YUV) e o vídeo reconstruído (YUV')

A principal diferença entre os processos previamente ilustrados é a transcodificação do arquivo YUV em H.264 nos processos 3 e 4. O objetivo dessa transcodificação é avaliar o quanto que a mesma irá impactar no momento em que o software FFMPEG decodificar de H.264 para YUV novamente.

Na avaliação de qualidade, foram utilizadas as métricas PSNR, SSIM e VQM previamente detalhadas no capítulo 2. A PSNR e a SSIM foram calculadas utilizando códigos de Matlab, enquanto que a métrica VQM foi calculada pelo software desenvolvido pelo instituto de pesquisa ITS (15). O cálculo da métrica VQM foi executado sem utilizar calibração espacial - considera que os quadros dos arquivos original e o analisado estão alinhados temporalmente - e utilizando o modelo de qualidade de vídeo *Developers Model*, que é um modelo bem aproximado e mais rápido em seis vezes do modelo padronizado pela ANSI (*NTIA General Model*).

5 RESULTADOS

5.1 SEQUÊNCIAS UTILIZADAS

As sequências utilizadas nas análises fazem parte de um banco de mídias para teste da coleção Derf (16) e estão sem compressão no formato YUV 4:2:0. Segue descrição e o primeiro frame de cada (figura 5.1).

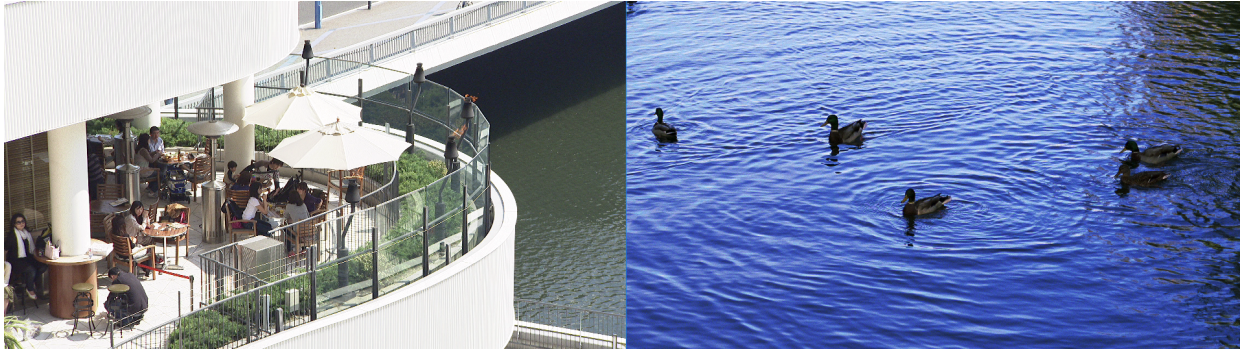


Figura 5.1: Primeiros quadros das sequências BQTerrace, à esquerda, e DucksTakeOff, à direita.

BQTerrace : Inicia com a vista aérea de uma terraço de um prédio, com três tochas acesas. Há um movimento de *pan* mostrando um rio, veículos e pessoas passando sobre uma ponte. O vídeo tem resolução de 1920x1080, 8 *bit-depth*, 60 quadros por segundos e duração de dez segundos.

DucksTakeOff : Vista área de um lago com 5 patos alçando voo, causando turbulência na água. O vídeo possui resolução de 1920x1080, 8 *bit-depth*, 50 quadros por segundos e duração de dez segundos.

5.2 ANÁLISE DOS RESULTADOS

Primeiramente, os resultados obtidos para os cenários mediante a utilização das métricas PSNR, SSIM e VQM serão ilustrados em forma de tabelas.

A análise comparativa entre cenários será feita sob a ótica da métrica VQM, visto que a mesma visa quantificar a qualidade subjetiva das mídias. Foram realizadas comparações entre os resultados obtidos nos cenários 1 com os resultados obtidos no cenário 3 e, de forma análoga, dos resultados obtidos no cenário 2 com os resultados obtidos no cenário 4. Desta maneira, será possível avaliar de forma mais clara os efeitos da transcodificação na qualidade do vídeo.

Por fim, será feita uma breve comparação da resposta do FFMPEG quando se analisa degradação espacial e temporal.

5.2.1 Cenário 1

Para o cenário 1, onde houve a degradação espacial e a reconstrução, obtivemos os resultados descritos na tabela 5.1. Como pode se verificar pelos resultados das métricas e pelas imagens de alguns frames de cada caso de destruição (figuras 5.2 e 5.3), a degradação espacial foi intensa em quase todos os casos, com exceção da redução da resolução para 960x540 e também pode-se verificar que a PSNR não demonstra essa degradação com a mesma intensidade que a SSIM e o VQM apresentam.

Sequência	Degradação espacial	PSNR (dB)	SSIM	VQM	si_loss	hv_loss	hv_gain	ati_gain	ati_loss
BQTerrace	1/5 = 384x216 pixels	20.6706	0.5400	0.5045	0.1577	0.1838	0.0770	0.0116	0.0743
	1/4 = 480x270 pixels	21.5644	0.5928	0.4093	0.1238	0.1468	0.0621	0.0088	0.067
	1/3 = 640x360 pixels	23.1162	0.6681	0.2512	0.0813	0.0662	0.0428	0.0056	0.0553
	1/2 = 960x540 pixels	26.7355	0.8111	0.1268	0.0357	0.0206	0.0256	0.0041	0.0407
DucksTakeOff	1/5 = 384x216 pixels	19.8552	0.4491	0.4427	0.1728	0.1301	0.0714	0.0142	0.0543
	1/4 = 480x270 pixels	20.9731	0.5393	0.3539	0.1349	0.1015	0.0591	0.0116	0.0469
	1/3 = 640x360 pixels	22.7671	0.6639	0.2342	0.0893	0.0538	0.0447	0.0092	0.0371
	1/2 = 960x540 pixels	26.6950	0.8419	0.1053	0.0376	0.0097	0.0262	0.0054	0.0264

Tabela 5.1: Resultados obtidos para o cenário 1

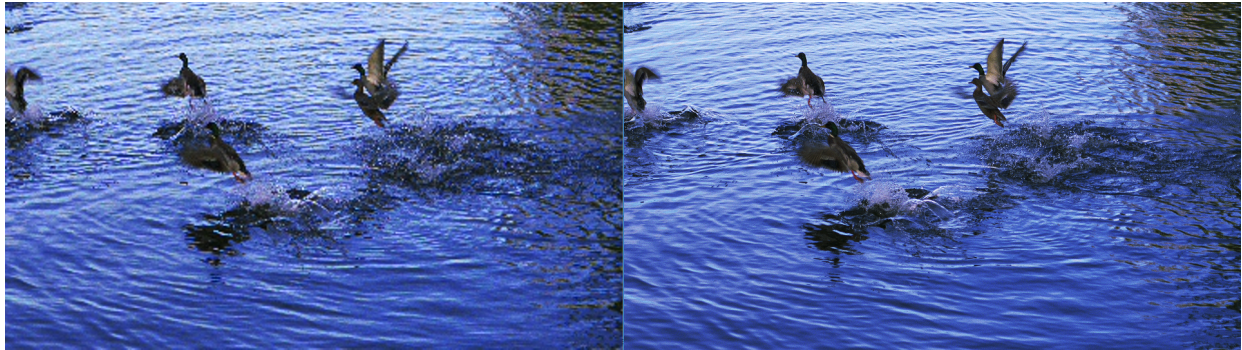


Figura 5.2: Centésimo frame da sequência DuckTakesOff original, à direita, e reconstruído, à esquerda para a degradação espacial de 384x216 pixels.

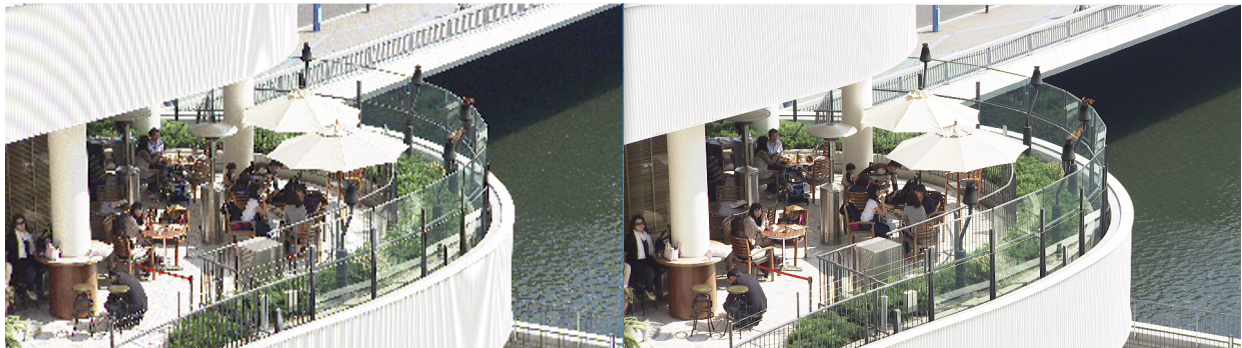


Figura 5.3: Primeiro frame da sequência BQTerrace original, à direita, e reconstruído, à esquerda para a degradação espacial de 384x216 pixels.

Ainda analisando as métricas e comparando as sequências, segundo a PSNR e a SSIM, a sequência DucksTakeOff apresentou valores inferiores para as duas métricas e portanto degradou-se mais que a outra sequência. Contudo, ao ver os resultados do VQM, em que valores inferiores significam uma qualidade maior, essa sequência obteve um melhor resultado.

5.2.2 Cenário 2

No cenário 2 houve a degradação temporal e reconstrução (*upsampling*). Neste cenário, foram realizados dois tipos de análise: A primeira análise é referente à replicação de frames para tratar a degradação temporal. Já a segunda análise é feita com o uso do filtro *minterpolate*, recurso presente nas bibliotecas do FFMPEG.

Este filtro implementa um algoritmo de predição que faz a conversão para um framerate especificado utilizando interpolação de movimento, a fim de recuperar as informações perdidas. Tal algoritmo possui compensação de movimento de bloco sobreposto e com estimação de movimento bidirecional (17). Os resultados se encontram nas tabelas 5.2 e 5.3.

Sequência	Degradação temporal	PSNR (dB)	SSIM	VQM	si_loss	hv_loss	hv_gain	ati_gain	ati_loss
BQTerrace	1/10 = 6 fps	18.6301	0.5601	0.4252	0.1126	0.1540	0.0980	0.0291	0.0316
	1/5 = 12 fps	20.7639	0.6500	0.3285	0.0870	0.1182	0.0712	0.0128	0.0392
	1/3 = 20 fps	22.9373	0.7378	0.2910	0.0725	0.1040	0.0611	0.0079	0.0455
	1/2 = 30 fps	25.5276	0.8212	0.1326	0.0359	0.0276	0.0302	0.0050	0.0339
DuckTakesOff	1/10 = 5 fps	17.6716	0.3582	0.6137	0.2008	0.1412	0.1021	0.1410	0.0285
	1/5 = 10 fps	21.1348	0.6287	0.4119	0.1410	0.1065	0.0734	0.0591	0.0320
	1/3 = 16 fps	25.5376	0.8279	0.2792	0.0867	0.0828	0.0484	0.0287	0.0326
	1/2 = 25 fps	27.7911	0.8916	0.1792	0.0532	0.0555	0.0337	0.0116	0.0251

Tabela 5.2: Resultados obtidos para o cenário 2 replicando frames

Sequência	Degradação temporal	PSNR (dB)	SSIM	VQM	si_loss	hv_loss	hv_gain	ati_gain	ati_loss
BQTerrace (com predição)	1/10 = 6 fps	25.8474	0.7788	0.2513	0.0745	0.0666	0.0518	0.0056	0.0529
	1/5 = 12 fps	30.6431	0.8726	0.0855	0.0233	0.0089	0.0246	0.0038	0.0249
	1/3 = 20 fps	31.8319	0.8970	0.0662	0.0149	0.0057	0.0212	0.0032	0.0212
	1/2 = 30 fps	35.8167	0.9353	0.0310	0.0003	0.0000	0.0140	0.0020	0.0148
DuckTakesOff (com predição)	1/10 = 5 fps	22.4012	0.6300	0.4425	0.1930	0.1080	0.0764	0.0073	0.0578
	1/5 = 10 fps	25.9717	0.8021	0.2563	0.0851	0.0767	0.0479	0.0051	0.0416
	1/3 = 16 fps	27.3696	0.8620	0.1960	0.0629	0.0570	0.0382	0.0051	0.0328
	1/2 = 25 fps	31.8405	0.9393	0.1170	0.0254	0.0361	0.0248	0.0031	0.0275

Tabela 5.3: Resultados obtidos para o cenário 2 com o algoritmo de predição

De forma análoga ao primeiro cenário, a PSNR obteve uma conclusão diferente da do VQM sobre qual sequência teve menos degradação, ou seja, a PSNR considerou que a DucksTakeOff teve mais degradação, independentemente da forma de reconstrução. Contudo, a SSIM no caso de uma degradação temporal, obteve as mesmas conclusões do VQM, ou seja, que a sequência BQTerrace teve uma menor degradação.

Analisando cada sequência individualmente, pode-se afirmar que a resolução temporal é mais afetada quando não é aplicado o algoritmo de predição de movimento e que o filtro do FFMPEG que o executa é eficaz, já que trouxe melhorias significativas em todas as



Figura 5.4: Frame 170 da sequência BQTerrace, à esquerda, e frame 171, à direita para a degradação temporal de 6fps sem algoritmo de previsão de movimento.

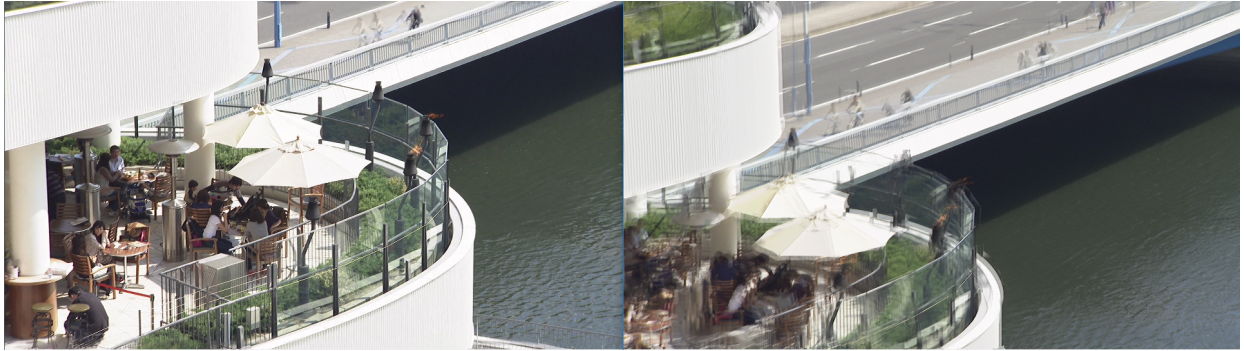


Figura 5.5: Frame 95 da sequência BQTerrace, à esquerda, e frame 175, à direita para a degradação temporal de 6fps com algoritmo de previsão.

métricas (vide tabelas 5.2 e 5.3, por exemplo), inclusive no pior caso em que a resolução foi de um décimo da original.

Já observando as mídias com a resolução de um décimo e um quinto da original, como na imagem 5.5, verifica-se a alta presença do efeito *ghost* nos casos em que foi aplicado o algoritmo de previsão. Este efeito incomoda alguns espectadores, mas pode ser mais agradável ao assistir do que ter a replicação de frames que termina com a sensação de sequência linear de acontecimentos de um vídeo, como pode ser observado no caso onde não houve previsão de movimento (figura 5.4). Esta última afirmação pode ser verificada também ao comparar os valores de VQM e suas submétricas.

5.2.3 Cenário 3

Sequência	Degradação espacial	PSNR (dB)	SSIM	VQM	si_loss	hv_loss	hv_gain	ati_gain	ati_loss
BQTerrace (com transcodificação)	1/5 = 384x216 pixels	20.9209	0.5497	0.5033	0.1582	0.1839	0.0767	0.0098	0.0747
	1/4 = 480x270 pixels	21.8731	0.6043	0.4064	0.1237	0.1458	0.0618	0.0067	0.0684
	1/3 = 640x360 pixels	23.4697	0.6791	0.2511	0.0817	0.0664	0.0429	0.0042	0.0559
	1/2 = 960x540 pixels	27.0915	0.8053	0.1307	0.0379	0.0225	0.0267	0.0023	0.0413
DuckTakesOff (com transcodificação)	1/5 = 384x216 pixels	19.9466	0.4521	0.4412	0.1743	0.1282	0.0712	0.0124	0.0551
	1/4 = 480x270 pixels	21.0961	0.5430	0.4412	0.1367	0.0997	0.0592	0.0094	0.0479
	1/3 = 640x360 pixels	22.8557	0.6625	0.2366	0.0924	0.0531	0.0450	0.0073	0.0388
	1/2 = 960x540 pixels	26.4033	0.8177	0.1257	0.0469	0.0155	0.0298	0.0037	0.0298

Tabela 5.4: Resultados obtidos para o cenário 3

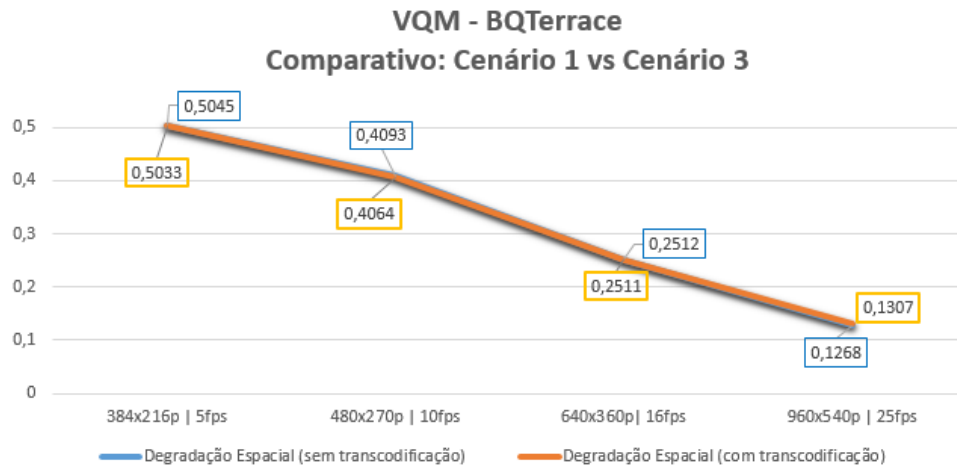


Figura 5.6: Comparação dos resultados obtidos nos cenários 1 e 3 para a sequência BQTerrace sob a ótica da métrica VQM

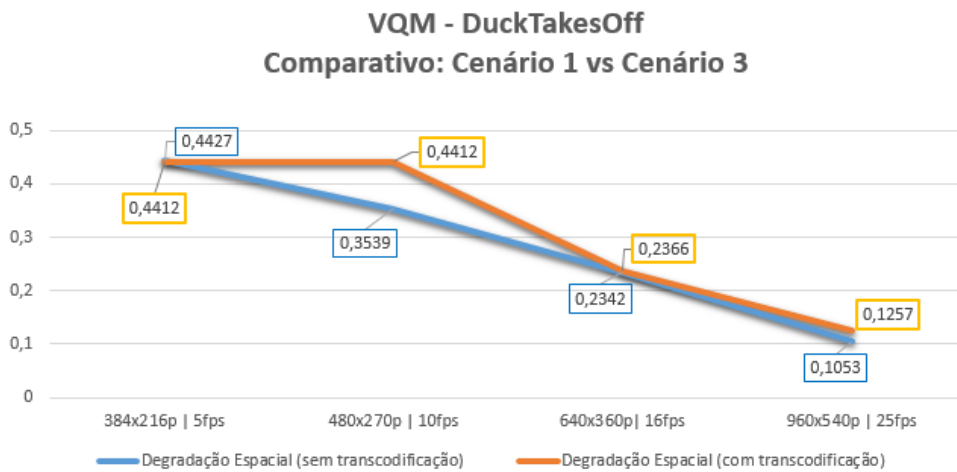


Figura 5.7: Comparação dos resultados obtidos nos cenários 1 e 3 para a sequência DucksTakeOff sob a ótica da métrica VQM

No cenário 3, para o arquivo YUV de entrada, houve a degradação espacial, codificação para H.264 e a transcodificação para o YUV'. Os resultados obtidos para o cenário em questão estão ilustrados na tabela 5.4.

Comparando-se os resultados do cenário 3 (tabela 5.4), com os resultados do cenário 1 (tabela 5.1), é possível notar alta semelhança nos valores encontrados, porém, em geral, houve uma piora nos valores para todas as métricas, isso deve-se ao algoritmo de transcodificação para H.264 feita pelo FFMPEG, com taxa de 50 Mbps.

As figuras 5.6 e 5.7 ilustram a semelhança entre os resultados encontrados nos cenários 1 e 3 para a métrica VQM. Constata-se que para o caso de degradação espacial a transcodificação para H.264, exceto para a sequência *DucksTakeOff* na degradação em 10 fps, não houve grande impacto na avaliação de qualidade.

5.2.4 Cenário 4

Para o cenário 4, de forma análoga ao cenário 2, foram realizados dois tipos de análise. A primeira delas é referente à replicação de frames e a segunda delas é com o uso do filtro *minterpolate* que implementa um algoritmo de predição de movimento. O intuito do cenário 4 é de se analisar o impacto da transcodificação das sequências. Tais resultados estão descritos nas tabelas 5.5 e 5.6.

Sequência	Degradação temporal	PSNR (dB)	SSIM	VQM	si_loss	hv_loss	hv_gain	ati_gain	ati_loss
BQTerrace (replicando frames)	1/10 = 6 fps	18.5603	0.5768	0.4069	0.1037	0.1531	0.0956	0.0242	0.0303
	1/5 = 12 fps	21.1807	0.6729	0.3150	0.0783	0.1177	0.0696	0.0100	0.0394
	1/3 = 20 fps	23.6606	0.7467	0.2780	0.0681	0.1009	0.0590	0.0063	0.0437
	1/2 = 30 fps	24.8040	0.7860	0.1409	0.0408	0.0275	0.0329	0.0045	0.0352
DuckTakesOff (replicando frames)	1/10 = 5 fps	18.4870	0.4366	0.5799	0.1993	0.1371	0.0938	0.1158	0.0339
	1/5 = 10 fps	22.8815	0.7190	0.3907	0.1414	0.1031	0.0678	0.0411	0.0372
	1/3 = 16 fps	26.0784	0.8342	0.2593	0.0819	0.0772	0.0480	0.0216	0.0307
	1/2 = 25 fps	27.2617	0.8623	0.1918	0.0597	0.0584	0.0369	0.0084	0.0284

Tabela 5.5: Resultados obtidos para o cenário 4 replicando frames

Sequência	Degradação temporal	PSNR (dB)	SSIM	VQM	si_loss	hv_loss	hv_gain	ati_gain	ati_loss
BQTerrace (com predição)	1/10 = 6 fps	25.4848	0.7776	0.2517	0.0765	0.0657	0.0524	0.0032	0.0539
	1/5 = 12 fps	30.0509	0.8590	0.0983	0.0288	0.0119	0.0271	0.0027	0.0278
	1/3 = 20 fps	30.8265	0.8683	0.0813	0.0214	0.0096	0.0247	0.0025	0.0232
	1/2 = 30 fps	33.4675	0.8863	0.0551	0.0101	0.0076	0.0193	0.0019	0.0162
DuckTakesOff (com predição)	1/10 = 5 fps	22.4135	0.6254	0.4455	0.1951	0.1090	0.0769	0.0055	0.0591
	1/5 = 10 fps	25.7180	0.7821	0.2670	0.0911	0.0782	0.0502	0.0033	0.0440
	1/3 = 16 fps	26.9960	0.8377	0.2092	0.0686	0.0598	0.0408	0.0040	0.0361
	1/2 = 25 fps	29.9356	0.8962	0.1432	0.0377	0.0417	0.0308	0.0032	0.0299

Tabela 5.6: Resultados obtidos para o cenário 4 com o algoritmo de predição

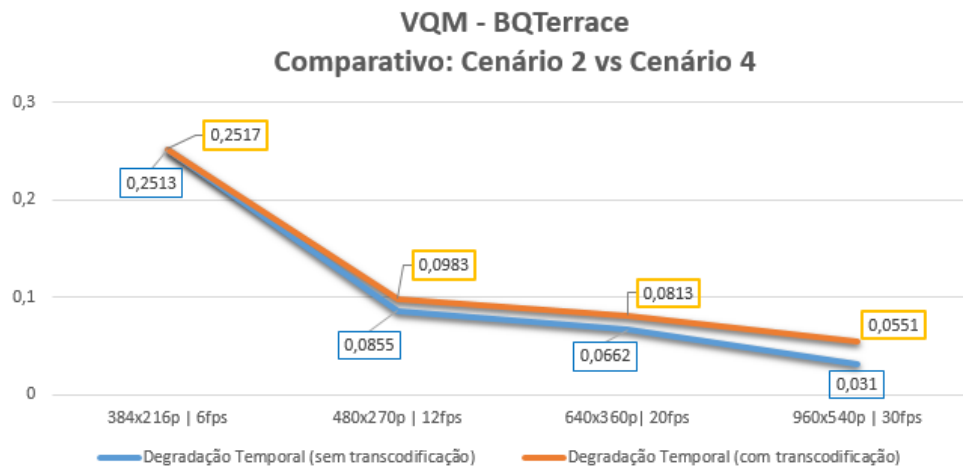


Figura 5.8: VQM: Comparativo dos cenários 2 e 4 para a sequência BQTerrace

Conforme já observado ao se analisar o cenário 2, a resolução temporal é mais afetada quando não é aplicado o algoritmo de predição de movimento e que o filtro do FFMPEG que o executa é eficaz. As figuras 5.8 e 5.9 ilustram a semelhança entre os resultados obtidos para a degradação temporal referente ao cenário 2 (sem transcodificação) e os resultados obtidos no cenário 4 (onde houve a transcodificação) considerando o uso do algoritmo de predição

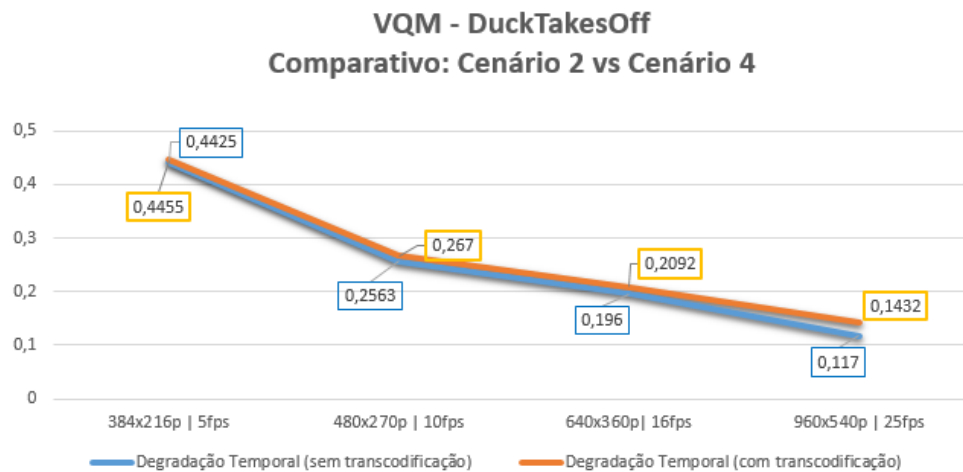


Figura 5.9: VQM: Comparativo dos cenários 2 e 4 para a sequência DucksTakeOff

de movimento. Nota-se que há semelhança entre os resultados, porém, a transcodificação impacta negativamente na avaliação de qualidade das sequências.

A seção a seguir trará explicações mais detalhadas sobre um ponto importante observado. Se trata da diferença ao se comparar o uso do algoritmo de predição de movimento com o uso da replicação frames. As figuras 5.10 e 5.11 ilustram a grande melhora de resultado ao se utilizar o algoritmo de predição de movimento.

5.3 COMPARATIVO ENTRE DEGRADAÇÃO ESPACIAL E TEMPORAL

As figuras 5.10 e 5.11 ilustram a resposta do FFMPEG em face às degradações temporais e espaciais. Foi levado em consideração a degradação temporal e espacial para os casos dos cenários 1 e 2 (onde não houve transcodificação) com o intuito de se observar apenas o tipo de resposta do software para duas mídias com características diferentes, e não comparar os efeitos da transcodificação.

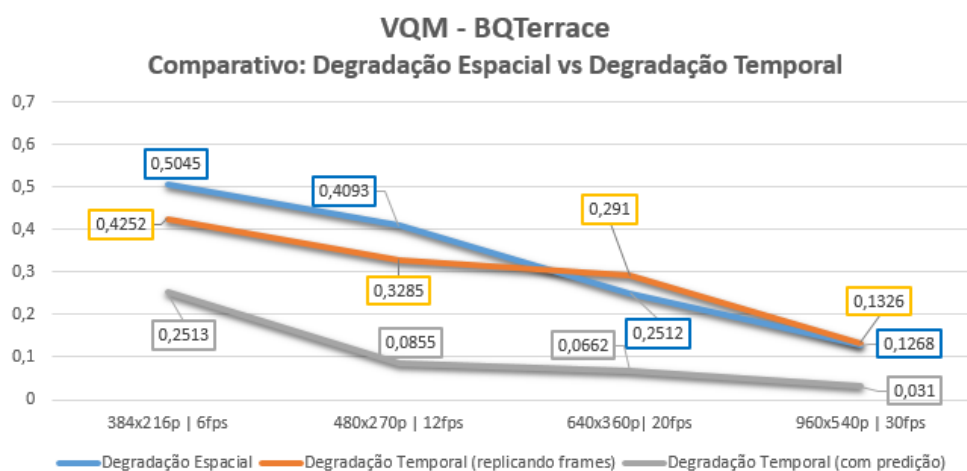


Figura 5.10: FFMPEG: Comparativo entre degradação espacial e temporal - BQTerrace

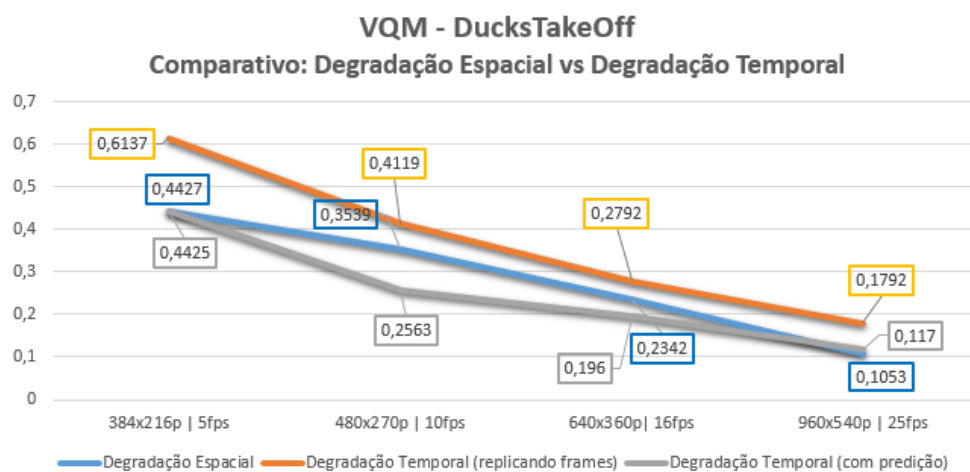


Figura 5.11: FFMPEG: Comparativo entre degradação espacial e temporal - DucksTakeOff

6 CONCLUSÃO

O presente trabalho teve como foco a avaliação do poder de transcodificação do software de código aberto FFMPEG, definindo cenários de aplicação que simulassem o ambiente de uma emissora de TV e, por meio do uso de métricas objetivas de avaliação de qualidade, identificar os pontos fortes e fracos deste software ao analisar duas sequências de vídeos distintas.

Ao se avaliar os resultados obtidos na análise dos quatro cenários implementados, observou-se que a transcodificação de H.264 do FFMPEG é eficiente ao tratar degradações temporais e espaciais (especialmente as temporais), obtendo resultados similares aos que são obtidos quando a reconstrução é feita diretamente com o arquivo descomprimido YUV.

Além disso, através da métrica VQM que tenta quantizar uma experiência subjetiva, foi possível concluir que o uso de algoritmos de predição de movimento faz-se necessário para melhorar consideravelmente a visualização da mídia pelo telespectador.

Através da análise das figuras 5.10 e 5.11 e das características das sequências testadas neste trabalho, é possível constatar que o FFMPEG trata melhor degradações temporais considerando que exista a predição de movimento, especialmente no caso de mídias onde há movimento de *pan* (como é o caso da sequência BQTerrace).

Desconsiderando o emprego do filtro *minterpolate*, nota-se que o FFMPEG trata melhor os casos de degradação temporal para mídias com características semelhantes à BQTerrace, ou seja, com muitas alterações espaciais no tempo - como o movimento de *pan* e *tilt* - especialmente nos piores casos. Quando se tem mídias semelhantes à DucksTakeOff - câmera estática e alterações espaciais pontuais - trata melhor os casos de degradação espacial.

Uma solução para tratar casos de alta degradação espacial é fazer uso de recursos como o videografismo para reproduzir as mídias degradadas, inserindo uma borda para que a reconstrução espacial seja menor e melhorando assim a experiência do telespectador. Tal solução vem sendo bastante usada em emissoras de TV atualmente.

Possíveis trabalhos futuros podem ter como iniciativa implementar a ideia de se refazer os testes munindo-se de uma quantidade maior de mídias, bem como a utilização de outras métricas de avaliação da qualidade e considerando também o caso de se incluir a análise subjetivas com pessoas reais. Adicionalmente, se torna válida a ideia de utilizar métricas sem referência. Desta forma, poderá obter uma análise mais ampla e completa acerca da ferramenta FFMPEG.

REFERÊNCIAS BIBLIOGRÁFICAS

- 1 AHMAD XIAOHUI WEI, Y. S. I.; ZHANG, Y.-Q. Video transcoding: An overview of various techniques and research issues. *IEEE TRANSACTIONS ON MULTIMEDIA*, v. 7, p. 793–800, 2005.
- 2 GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing*. [S.l.: s.n.], 2006.
- 3 TEKALP, A. M. *Digital Video Processing*. [S.l.: s.n.], 1995.
- 4 BOVIK, A. C. *Handbook of Image and Video Processing (Communications, Networking and Multimedia)*. [S.l.: s.n.], 2005.
- 5 BOBICK, A. *CS 4495 Computer Vision Motion and Optic Flow*. [S.l.], 2009. Disponível em: <<https://www.cc.gatech.edu/~afb/classes/CS4495-Fall2014/slides/CS4495-OpticFlow.pdf>>.
- 6 WIEN, M. *High Efficiency Video Coding – Coding Tools and Specification*. [S.l.: s.n.], 2015.
- 7 SILVA, E. P. F. da. Advanced heterogeneous video transcoding. *School of Electronic Engineering and Computer Science, University of London*, v. 1, p. 212–218, 2012.
- 8 XIN, C.-W. J.; MING-TING. Digital video transcoding. *PROCEEDINGS OF THE IEEE*, v. 93, p. 84–90, 2005.
- 9 GOBO, A. L. B. Ferramenta para criação de transport stream com múltiplos programas para o padrão brasileiro de televisão digital. *UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL*, v. 1, p. 18–20, 2016.
- 10 RICHARDSON, I. E. *The H.264 Advanced Video Compression Standard*. [S.l.: s.n.], 2003.
- 11 KIM, D. P. S.; LEE, S. Ssim-based distortion metric for film grain noise in hevc. *Yonsei University, Seoul, Republic of Korea*, v. 1, p. 490–492, 2017.
- 12 WANG A. C. BOVIK, H. R. S. Z.; SIMONCELLI, E. P. Image quality assessment: From error visibility to structural similarity. *EEE Transactions on Image Processing*, v. 13, p. 600–612, 2004.
- 13 JUNG, A. Comparison of video quality assessment methods. *Blekinge Institute of Technology*, v. 1, p. 24–27, 2017.
- 14 PINSON, M. H.; WOLF, S. A new standardized method for objectively measuring video quality. *EEE Transactions on Image Processing*, v. 1, p. 1–12, 2003.
- 15 INSTITUTE FOR TELECOMMUNICATION SCIENCES. *Description of Video Quality Metric (VQM) Software*. 2018. Disponível em: <<https://www.its.bldrdoc.gov/resources/video-quality-research/guides-and-tutorials/description-of-vqm-tools.aspx>>. Acesso em: 23 jun. 2018.
- 16 DERF’S Test Media Collection. 2018. Disponível em: <<https://media.xiph.org/video/derf/>>. Acesso em: 02 abr. 2018.
- 17 FFMPEG Documentation. 2018. Disponível em: <<https://www.ffmpeg.org/ffmpeg-all.html#minterpolate>>. Acesso em: 02 jun. 2018.

APÊNDICES

Mídia	Cenário	Tipo de degradação	Degradação	PSNR (dB)	SSIM	VQM	si_loss	hv_loss	hv_gain	ati_gain	ati_loss
BQTerrace	1	Espacial (sem transcodificação)	384x216	20.6706	0.5400	0.5045	0.1577	0.1838	0.0770	0.0116	0.0743
			480x270	21.5644	0.5928	0.4093	0.1238	0.1468	0.0621	0.0088	0.0678
			640x360	23.1162	0.6681	0.2512	0.0813	0.0662	0.0428	0.0056	0.0553
			960x540	26.7355	0.8111	0.1268	0.0357	0.0206	0.0256	0.0041	0.0407
	3	Espacial (com transcodificação)	384x216	20.9209	0.5497	0.5033	0.1582	0.1839	0.0767	0.0098	0.0747
			480x270	21.8731	0.6043	0.4064	0.1237	0.1458	0.0618	0.0067	0.0684
			640x360	23.4697	0.6791	0.2511	0.0817	0.0664	0.0429	0.0042	0.0559
			960x540	27.0915	0.8053	0.1307	0.0379	0.0225	0.0267	0.0023	0.0413
	2	Temporal - replicando frames (sem transcodificação)	6fps	18.6301	0.5601	0.4252	0.1126	0.1540	0.0980	0.0291	0.0316
			12fps	20.7639	0.6500	0.3285	0.0870	0.1182	0.0712	0.0128	0.0392
			20fps	22.9373	0.7378	0.2910	0.0725	0.1040	0.0611	0.0079	0.0455
			30fps	25.5276	0.8212	0.1326	0.0359	0.0276	0.0302	0.0050	0.0339
	4	Temporal - replicando frames (com transcodificação)	6fps	18.5603	0.5768	0.4069	0.1037	0.1531	0.0956	0.0242	0.0303
			12fps	21.1807	0.6729	0.3150	0.0783	0.1177	0.0696	0.0100	0.0394
			20fps	23.6606	0.7467	0.2780	0.0681	0.1009	0.0590	0.0063	0.0437
			30fps	24.8040	0.7860	0.1409	0.0408	0.0275	0.0329	0.0045	0.0352
	2	Temporal - com predição (sem transcodificação)	6fps	25.8474	0.7788	0.2513	0.0745	0.0666	0.0518	0.0056	0.0529
			12fps	30.6431	0.8726	0.0855	0.0233	0.0089	0.0246	0.0038	0.0249
			20fps	31.8319	0.8970	0.0662	0.0149	0.0057	0.0212	0.0032	0.0212
			30fps	35.8167	0.9353	0.0310	0.0003	0.0000	0.0140	0.0020	0.0148
	4	Temporal - com predição (com transcodificação)	6fps	25.4848	0.7776	0.2517	0.0765	0.0657	0.0524	0.0032	0.0539
			12fps	30.0509	0.8590	0.0983	0.0288	0.0119	0.0271	0.0027	0.0278
			20fps	30.8265	0.8683	0.0813	0.0214	0.0096	0.0247	0.0025	0.0232
			30fps	33.4675	0.8863	0.0551	0.0101	0.0076	0.0193	0.0019	0.0162

Tabela 1: Dados obtidos para a sequência BQTerrace

Mídia	Cenário	Tipo de degradação	Degradação	PSNR (dB)	SSIM	VQM	si_loss	hv_loss	hv_gain	ati_gain	ati_loss
DucksTakeOff	1	Espacial (sem transcodificação)	384x216p	19.8552	0.4491	0.4427	0.1728	0.1301	0.0714	0.0142	0.0543
			480x270p	20.9731	0.5393	0.3539	0.1349	0.1015	0.0591	0.0116	0.0469
			640x360p	22.7671	0.6639	0.2342	0.0893	0.0538	0.0447	0.0092	0.0371
			960x540p	26.6950	0.8419	0.1053	0.0376	0.0097	0.0262	0.0054	0.0264
	3	Espacial (com transcodificação)	384x216p	19.9466	0.4521	0.4412	0.1743	0.1282	0.0712	0.0124	0.0551
			480x270p	21.0961	0.5430	0.4412	0.1367	0.0997	0.0592	0.0094	0.0479
			640x360p	22.8557	0.6625	0.2366	0.0924	0.0531	0.0450	0.0073	0.0388
			960x540p	26.4033	0.8177	0.1257	0.0469	0.0155	0.0298	0.0037	0.0298
	2	Temporal - replicando frames (sem transcodificação)	5fps	17.6716	0.3582	0.6137	0.2008	0.1412	0.1021	0.1410	0.0285
			10fps	21.1348	0.6287	0.4119	0.1410	0.1065	0.0734	0.0591	0.0320
			16fps	25.5376	0.8279	0.2792	0.0867	0.0828	0.0484	0.0287	0.0326
			25fps	27.7911	0.8916	0.1792	0.0532	0.0555	0.0337	0.0116	0.0251
	4	Temporal - replicando frames (com transcodificação)	5fps	18.4870	0.4366	0.5799	0.1993	0.1371	0.0938	0.1158	0.0339
			10fps	22.8815	0.7190	0.3907	0.1414	0.1031	0.0678	0.0411	0.0372
			16fps	26.0784	0.8342	0.2593	0.0819	0.0772	0.0480	0.0216	0.0307
			25fps	27.2617	0.8623	0.1918	0.0597	0.0584	0.0369	0.0084	0.0284
	2	Temporal - com predição (sem transcodificação)	5fps	22.4012	0.6300	0.4425	0.1930	0.1080	0.0764	0.0073	0.0578
			10fps	25.9717	0.8021	0.2563	0.0851	0.0767	0.0479	0.0051	0.0416
			16fps	27.3696	0.8620	0.1960	0.0629	0.0570	0.0382	0.0051	0.0328
			25fps	31.8405	0.9393	0.1170	0.0254	0.0361	0.0248	0.0031	0.0275
	4	Temporal - com predição (com transcodificação)	5fps	22.4135	0.6254	0.4455	0.1951	0.1090	0.0769	0.0055	0.0591
			10fps	25.7180	0.7821	0.2670	0.0911	0.0782	0.0502	0.0033	0.0440
			16fps	26.9960	0.8377	0.2092	0.0686	0.0598	0.0408	0.0040	0.0361
			25fps	29.9356	0.8962	0.1432	0.0377	0.0417	0.0308	0.0032	0.0299

Tabela 2: Dados obtidos para a sequência DuckTakesOff