

**UNIVERSIDADE DO EXTREMO SUL CATARINENSE - UNESC**

**CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**LAURO AGUIRRE NASCIMENTO**

**ANÁLISE COMPARATIVA DE DESEMPENHO ENTRE OS CODIFICADORES DE  
VÍDEO H.264/AVC JM E H.264/SVC JSVM**

**CRICIÚMA**

**2013**

**LAURO AGUIRRE NASCIMENTO**

**ANÁLISE COMPARATIVA DE DESEMPENHO ENTRE OS CODIFICADORES DE  
VÍDEO H.264/AVC JM E H.264/SVC JSVM**

Trabalho de Conclusão de Curso, apresentado para obtenção do grau de bacharel no curso de Ciência da Computação da Universidade do Extremo Sul Catarinense, UNESC.

Orientador: Prof. MEng. Evânio Ramos Nicoleit

**CRICIÚMA**

**2013**

LAURO AGUIRRE NASCIMENTO


**ANÁLISE COMPARATIVA DE DESEMPENHO ENTRE OS CODIFICADORES  
DE VÍDEO H.264/AVC JM E H.264/SVC JSVM**

Trabalho de Conclusão de Curso  
aprovado pela Banca Examinadora para  
obtenção do Grau de Bacharel, no Curso  
de Ciência da Computação da  
Universidade do Extremo Sul  
Catarinense, UNESC, com Linha de  
Pesquisa em codificação de vídeo.

Criciúma, 25 de junho de 2013.

**BANCA EXAMINADORA**

  
Prof. MEng. Evânio Ramos Nicoleit - UNESC - Orientador

  
Prof. MSc. Paulo João Martins - UNESC

  
Prof. MSc. Luciano Antunes - UNESC

## **AGRADECIMENTOS**

Agradeço a todos que direta ou indiretamente me apoiaram e deram forças diante o desenvolvimento de todo o trabalho, mesmo quando eu estava prestes a desistir acreditando que o tempo não estava do meu lado.

Em especial as pessoas mais próximas de mim, que aturaram meus mau humores, reinos e caras fechadas, e que nestes momentos souberam entender e relevar tudo com tranquilidade.

Muito obrigado a todos!

**“Só lutamos por aquilo que amamos, só amamos aquilo que respeitamos e só respeitamos aquilo que conhecemos.”**

**Adolf Hitler**

## RESUMO

Vídeos digitais estão presentes em praticamente todos os ambientes do nosso dia-a-dia. Vivemos em uma realidade onde imagens são gravadas e transmitidas a todo o momento por diversos dispositivos com diferentes tecnologias. A diversidade de dispositivos, e a diferença de capacidade de cada um gera um grande desafio quando se trata de obter bom desempenho em cada local em que um vídeo é exibido. Vídeos de alta qualidade requerem aparelhos mais potentes, impossibilitando sua exibição em aparelhos mais modestos. Já vídeos de menor qualidade poderão ser exibidos em uma quantidade mais significativa de aparelhos, porém não estará possibilitando a utilização de toda a capacidade de estruturas de exibição mais potentes. Diversas tecnologias para codificação e transmissão de vídeos digitais são desenvolvidas e aperfeiçoadas. Dentre estas tecnologias, destaca-se o padrão H.264, que é atualmente o estado da arte com relação à codificação de vídeo. Este padrão divide-se em H.264/AVC e H.264/SVC. O padrão H.264/SVC foi desenvolvido a partir do H.264/AVC, aperfeiçoando seus algoritmos de compensação e estimação de movimentos e incorporando suporte a escalabilidade. Este trabalho tem o propósito de realizar uma análise comparativa entre os dois padrões com seus softwares de referência. Para isto, foram codificadas diferentes sequências de vídeo para diferentes níveis de qualidade. A análise consistiu em confrontar as métricas geradas por cada codificador para que fosse possível identificar em quais pontos cada codificador se destaca. Os resultados demonstram que o software do codificador com suporte a escalabilidade é mais eficiente que o codificador sem suporte a escalabilidade, esclarecendo assim melhorias implementadas no codificador JSVM além do suporte a escalabilidade.

**Palavras-chave:** Codificação de vídeo. H.264/AVC. H.264/SVC. Escalabilidade.

## ABSTRACT

Digital videos are present in every ambient everyday. We live in a reality where images are recorded and transmitted all the time by many dispositives with different technologies. The diversity of devices and the difference of limitations of each make a great challenge about obtaining a good performance over each situation where the videos are displayed. High quality videos require more powerful equipment, making it impossible to view in simpler devices. However, lower quality video may be displayed in a more significant amount of devices, but is not enabling to use all the capacity of powerful structures. Several technologies for encoding and transmission of digital videos are developed and refined. Among these technologies, stands the H264 video coding standard, which is a state-of-the-art in video coding. This standard is divided in H.264/AVC and H.264/SVC. The H.264/SVC standard was developed from H.264/AVC, perfecting the algorithms and motion estimation algorithms and incorporating scalability support. This paper aims to conduct a comparative performance analysis of two standards with their software reference. To this, different video sequences were encoded in different quality levels. The analysis consist in confront the values generated by each encoder and identify which points each encoder excels. The result shows the software that supports scalability encoder is more efficient then the encoder without scalability support, this clarifying improvements implemented in JSVM encoder addition to scalability support.

**Keywords:** Video Coding. H.264/AVC. H.264/SVC. Scalability.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Quadros de imagens.....	14
Figura 2 – Divisão de macroblocos em uma imagem.....	17
Figura 3 – Imagem monocromática com 256 níveis de cinza.....	19
Figura 4 – Imagem monocromática após redução para 16 níveis de cinza. ....	19
Figura 5 – Diagrama de quadros de um codificador de vídeo.....	20
Figura 6 – Perfis de codificação do padrão H.264/AVC. ....	25
Figura 7 – Diagrama de quadros do Padrão H.264/AVC. ....	27
Figura 8 – Interpolação para $\frac{1}{2}$ de pixel.....	30
Figura 9 – Interpolação para $\frac{1}{4}$ de pixel.....	31
Figura 10 – Comando de chamada do codificador.....	38
Figura 11 – Apresentação de frames codificados em tempo real.....	41
Figura 12 – Resultados ao término da codificação.....	42
Figura 13 – Chamada do decodificador da JSVM.....	42
Figura 14 – Chamada do codificador da JM.....	43
Figura 15 – Tabela de resultados em tempo real do codificador JM.....	45
Figura 16 – Média dos resultados gerados pelo codificador JM.....	46
Figura 17 – Sequências de vídeo.....	47
Figura 18 – Análise de Qualidade x Quantização. ....	50
Figura 19 – Análise de Complexidade Computacional x Quantização.....	51
Figura 20 – Análise de Taxa de bits x Qualidade.....	52
Figura 21 – Análise de Complexidade x Taxa de Bits.....	53
Figura 22 – Análise de qualidade x taxa de <i>bits</i> com suporte a escalabilidade.....	54
Figura 23 – Análise de complexidade x taxa de <i>bits</i> com suporte a escalabilidade. .	54

## LISTA DE TABELAS

Tabela 1 – Comparação entre as diferentes características dos padrões H.264/AVC, MPEG-4 Parte 2 e MPEG-2. ....	23
Tabela 2 – Comparativo da taxa de <i>bits</i> em relação ao padrão H.264/AVC. ....	23
Tabela 3 – Arquivo principal de configuração da JSVM. ....	39
Tabela 4 – Arquivo de configuração do <i>layer 0</i> . ....	39
Tabela 5 – Arquivo de configuração do <i>layer 1</i> . ....	39
Tabela 6 – Descrição das colunas da Figura 11. ....	41
Tabela 7 – Parâmetros do codificador da JM. ....	43
Tabela 8 – Principais campos do arquivo de configuração. ....	45
Tabela 9 – Colunas dos resultados apresentados pelo codificador da JM. ....	46

## LISTA DE ABREVIATURAS E SIGLAS

CABAC	Codificação aritmética adaptativa ao contexto
CAVLC	Codificação de entropia com códigos de comprimento de palavra variável adaptativas ao contexto
DCT	Transformada discreta do cosseno
FPS	<i>Frames</i> por segundo
FRExt	<i>Fidelity Range Extensions</i>
HD	<i>High Definition</i>
HW	<i>Hardware</i>
JVT	<i>Joint Video Team</i>
JSVM	<i>Joint Scalable Video Model</i>
MC	<i>Motion Compensation</i> (Compensação de Movimento)
ME	<i>Motion Estimation</i> (Estimação de Movimento)
MPEG	<i>ISO/IEC Moving Picture Experts Group</i>
NAL	Camada de abstração de rede
QP	Taxa de quantização
SO	Sistema Operacional
SW	<i>Software</i>
SVC	<i>Scalable Video Coding</i>
VCEG	<i>ITU-T Video Coding Experts Group</i>
VCL	Camada de codificação de vídeo

## SUMÁRIO

<b>1 Introdução</b> .....	<b>11</b>
1.1 Objetivo Geral .....	11
1.2 Objetivos Específicos .....	11
1.3 Justificativa.....	12
1.4 Estrutura do Projeto .....	12
<b>2 Vídeo Digital</b> .....	<b>14</b>
2.1 Compressão de Vídeo Digital.....	15
2.2 Redundância Temporal e Espacial.....	16
2.3 Redundância em Entropia.....	17
2.4 Redundância Psico-visual .....	18
2.5 Codificador/Decodificador de Vídeo .....	20
<b>3 O Padrão H.264/AVC</b> .....	<b>22</b>
3.1 Perfis e Níveis .....	24
3.2 Formato dos Dados Codificados .....	26
<b>4 Núcleo do Padrão H.264/AVC</b> .....	<b>27</b>
4.1 Módulo de Estimação de Movimento .....	29
4.2 Módulo de Compensação de Movimento .....	32
4.3 O Padrão H.264/SVC .....	33
<b>5 Trabalhos Correlatos</b> .....	<b>35</b>
5.1 Avaliação De Desempenho Da Escalabilidade Espacial Na Recomendação H.264/SVC Para Codificação de Vídeo.....	35
5.2 Performance Analysis of SVC .....	35
5.3 Comparison of Stereo Video Coding Support in MPEG-4 MAC, H.264/AVC and H.264/SVC .....	35
5.4 Um estudo Sobre o Padrão H.264/AVC de Compressão de Vídeo.....	36
<b>6 Comparação dos Codificadores</b> .....	<b>37</b>
6.1 Metodologia.....	37
<b>6.1.1 Joint Scalable Video Model (JSVM)</b> .....	<b>37</b>
<b>6.1.2 Codificação de vídeo com JM</b> .....	<b>42</b>
<b>6.1.3 Sequências de vídeo utilizadas para testes</b> .....	<b>46</b>
6.2 Resultados .....	48
<b>6.2.1 Análise Comparativa de Qualidade x Quantização</b> .....	<b>50</b>
<b>6.2.2 Taxa de Quantização x Complexidade</b> .....	<b>50</b>

<b>6.2.3 Análise de Taxa de Bits x Qualidade .....</b>	<b>51</b>
<b>6.2.4 Análise da Complexidade x Taxa de Bits .....</b>	<b>52</b>
<b>6.2.5 Análise do Resultado de Codificação com Escalabilidade .....</b>	<b>53</b>
<b>7 Conclusão .....</b>	<b>55</b>

## **1 Introdução**

A transmissão de vídeo está cada dia mais presente no cotidiano de todos. Na televisão, em filmes, em exibições pela Internet e até mesmo em conversas por celular, o vídeo vem conquistando seu espaço dentre os principais meios de comunicação da atualidade.

Com as novas tecnologias de imagens digitais, cada vez mais se faz necessária uma solução capaz de permitir a transmissão de imagens em alta definição através de sinais limitados. Atualmente, enviar e receber imagens HD através de dispositivos móveis, por exemplo, tem se tornado uma prática comum dentre os amantes da tecnologia. Porém, para que seja possível esta transferência sem que sejam apresentadas imagens travadas ou distorcidas enfrenta-se um grave problema: imagens de alta definição exigem uma grande quantidade de informação para serem transferidas, e isto exige uma taxa de transferência inviável a nível de HW, restando assim buscar uma solução de SW capaz de reduzir a quantidade de informação a ser transmitida.

É neste contexto que esta pesquisa se enquadra, realizando uma análise comparativa entre dois modos de compressão de vídeo (H.264/AVC JM e H.264/SVC JSVM) que representam o estado da arte na área de codificação de vídeo da atualidade, a fim de obtermos informações que definam em que pontos é possível obter um maior desempenho referente a qualidade de imagem, taxa de bits gerada e complexidade computacional a nível de tempo de execução necessário dentre os codificadores.

### **1.1 OBJETIVO GERAL**

Analisar comparativamente o desempenho entre os codificadores de vídeo H.264/AVC JM e H.264/SVC JSVM.

### **1.2 OBJETIVOS ESPECÍFICOS**

Este projeto de pesquisa está sendo desenvolvido com os seguintes intuitos:

- a) Descrever os fundamentos de codificação de vídeo;
- b) Apresentar os princípios de escalabilidade de vídeo;
- c) Avaliar comparativamente o processo de codificação de vídeo baseado nas Recomendações H.264/AVC através da JM e H.264/SVC através da JSVM;

### 1.3 JUSTIFICATIVA

Os padrões H.264/AVC e H.264/SVC representam o estado-da-arte em codificação de vídeo. Contudo, os padrões especificam os requerimentos de compatibilidade de decodificação, deixando aberto possíveis aprimoramentos que dizem respeito à forma de realização da codificação, que podem ser alcançados a fim de torná-lo mais eficiente.

Os sistemas de codificação de vídeo dependem basicamente de quatro características fundamentais: escala abordada, qualidade do vídeo resultante, taxa de bits associada e complexidade computacional envolvida.

Sabe-se que os SW de referência disponibilizados pelo JVT auxiliam no desenvolvimento de sistemas construídos com HW específicos dedicados e programados diretamente por meio de mnemônicos. Entretanto, não servem para a implementação diretamente, pois são construídas sobre linguagens de alto nível tais como C++, dentre outras. Contudo, servem para simular os processos de codificação e decodificação, perdendo para os sistemas que operam em tempo real somente na questão de complexidade computacional ou propriamente no tempo envolvido.

Esta pesquisa busca analisar comparativamente os codificadores de vídeo H.264/AVC JM e H.264/SVC JSVM, visando evidenciar em que características e parâmetros obtém-se melhor resultado para cada estrutura de codificação.

### 1.4 ESTRUTURA DO PROJETO

No Capítulo 1, é apresentada a introdução, os objetivos e a justificativa deste projeto de pesquisa.

O Capítulo 2 aborda os conceitos de vídeo digital. Neste capítulo também são trazidos conceitos de codificação/decodificação de vídeo, compressão de vídeo e técnicas de compressão.

O Capítulo 3 contém a definição do funcionamento do codificador H.264/AVC. Traz também a descrição dos perfis e níveis de codificação suportados pelo codificador.

O Capítulo 4 apresenta o funcionamento do núcleo do codificador H.264, com os módulos de compensação e estimação de movimento. Neste capítulo também conta a descrição do modelo H.264/SVC, identificando a incorporação de suporte a escalabilidade e conceituando escalabilidade de vídeo.

O Capítulo 5 apresenta uma breve descrição de alguns trabalhos que possuem relação com o foco desta pesquisa.

No Capítulo 6 é descrita a metodologia utilizada para a extração e análise das informações e os resultados obtidos durante os testes.

O Capítulo 7 apresenta a conclusão obtida por meio da análise dos resultados descritos no Capítulo 6.

## 2 Vídeo Digital

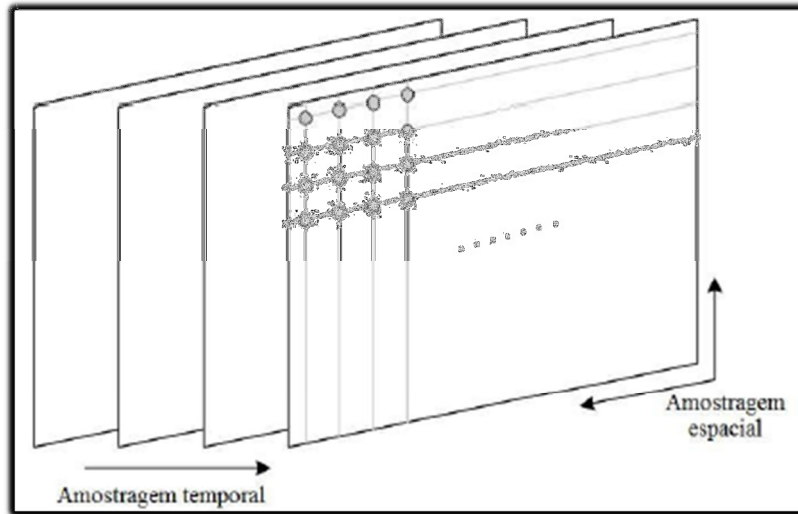
Bordignon (2000) afirma que a sensação de animação em uma imagem se dá pela propriedade da retina do olho humano de reter uma imagem por alguns milissegundos até que esta se perca. Com isto, a exibição de diversas imagens com pouca alteração de posição dos pixels faz com que o cérebro interprete aquela sequência de imagens como movimento.

Bordignon (2000), diz que uma imagem é um objeto bidimensional, cuja sua representação é feita através de mapeamento. Agregando as definições de Borignon (2000) e Souza (2010), é possível chegar a mesma conclusão descrita por Porto (2012), que define um vídeo digital como sendo uma sequência de imagens distintas, as quais tratarei nesta pesquisa como quadros, sendo captadas e exibidas em um determinado intervalo de tempo.

Em sua tese de mestrado, Souza (2010) explana que um vídeo digital pode ser representado como um sinal tridimensional, sendo uma dimensão de tempo e duas de espaço.

Como descrito anteriormente, a representação de uma imagem é feita através de mapeamento, que resulta na divisão da imagem em pontos de um plano cartesiano. Neste relato, estes pontos serão denominados *pixel*. A Figura 1 apresenta este sinal tridimensional, com a sequência dos quadros (Amostragem espacial) no espaço de tempo (Amostragem temporal) e a representação da disposição dos *pixels* em cada quadro.

Figura 1 – Quadros de imagens.



Fonte: Amorim (2009, p. 5).

## 2.1 COMPRESSÃO DE VÍDEO DIGITAL

Devido ao fato de a retina humana reter a imagem por cerca de 40 milissegundos, torna-se possível passar a impressão de movimento exibindo aproximadamente 25 quadros por segundo. Devido a isto, qualquer número de quadros a mais que sejam captados não irão representar nenhum ganho na qualidade do vídeo, tornando-se assim desnecessários (BORDIGNON, 2000).

Este é um exemplo de técnica para compressão de vídeo. Como descrito por Gonzalez, Woods (2003, p. 219), “o termo *compressão de dados* refere-se ao processo de redução da quantidade de dados necessária para representar certa quantidade de informação”, e, sem o emprego de técnicas de compressão de vídeo, hoje seria completamente inviável a transmissão de imagens em tempo real por qualquer que fosse o meio de comunicação. Soares (2007), em seu estudo sobre codificação de sinais, pode observar que se não for tratado, um vídeo preto e branco de cerca de 1 minuto pode gerar 600 *Mbytes*. Já Rosa (2007), através de seus estudos, estimou que a transmissão de um vídeo digital de televisão com definição normal (720x480 *pixels*), e duração de cerca de 10 minutos, teria um tamanho de aproximadamente 19 GB e ainda um vídeo com a mesma duração porém em HD (1920x1080) o tamanho passaria para 112 GB, o que tornaria difícil e custoso o armazenamento de tanta informação e ainda inviável sua transmissão em tempo real.

De acordo com Greco (2011), o princípio da compressão de vídeo é a

eliminação de redundâncias do sinal do mesmo. Castro (2011), em sua produção à respeito de codificação de sinais, salienta três tipos de redundâncias que podem ser trabalhadas para uma efetiva compressão de vídeo e que são utilizadas pelos codificadores H.264/AVC e H.264/SVC, que são o foco desta pesquisa: Redundância Temporal e Espacial, Redundância em Entropia e Redundância psico-visual.

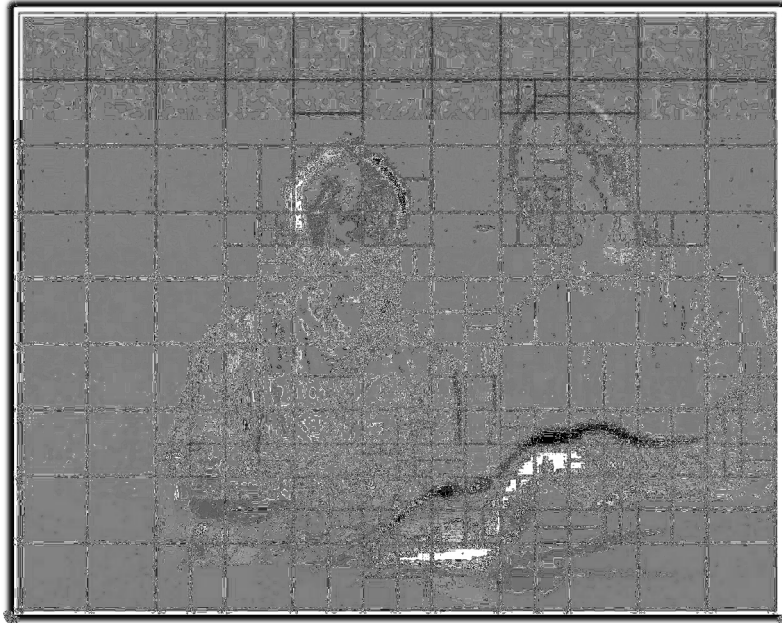
## 2.2 REDUNDÂNCIA TEMPORAL E ESPACIAL

Neste caso, são observados os quadros de imagens que compõem o vídeo e se identificam os sinais (*pixels*) com uma intensidade muito próxima em um mesmo quadro (redundância espacial) ou que se repetem em uma sequência de quadros (redundância temporal).

Em um vídeo digital, o espaço de cores comumente utilizado é o YCbCr, no qual Y representa a intensidade de brilho, Cb a intensidade da cor azul e Cr a intensidade da cor vermelha no pixel. Uma grande vantagem do YCbCr é que a informação de cor está completamente separada da informação de brilho, de modo que estas informações podem ser tratadas de forma independente (Rosa, 2007). Para a compressão de vídeo, este ponto facilita na identificação de redundâncias nas imagens.

Como citado por Castro (2011), um pixel é correlacionado com seus vizinhos, portanto, aproveitando-se esta propriedade, a compressão pode ser realizada através de uma previsão sobre um grupo de pixels, ao ponto em que se encontrem valores muito semelhantes em determinadas áreas dos quadros. No processo de codificação de vídeo, os pixels são separados em grupos denominados macroblocos, onde são extraídos os valores de Y, Cb e Cr para cada macrobloco, adotando-se assim um processo de codificação para cada conjunto de pixels ao invés de se tratar cada pixel separadamente. “Em geral, uma partição grande é apropriada para áreas mais homogêneas do quadro e uma partição menor tende a ser mais apropriada para áreas com muitos detalhes” (AGOSTINI, 2007, p. 48). Um exemplo desta divisão de quadros pode ser percebida logo a seguir, na Figura 2.

Figura 2 – Divisão de macroblocos em uma imagem.



Fonte: Silva (2007, p.14).

Para eliminação de redundância temporal, é avaliada a capacidade de distinção entre as imagens pela visão humana. A captação de imagens feita por uma câmera de vídeo pode chegar a centenas de imagens por segundo, porém para que se tenha um resultado satisfatório, a sucessão deve ficar entre 24 a 30 quadros por segundo para que o sistema visual humano perceba a sequência de quadros como um vídeo contínuo (Amorim, 2009). Com isto, a definição do intervalo de tempo presente entre os quadros a serem codificados no vídeo permite que as demais imagens presentes nestes intervalos de tempo que não venham a ser processadas possam ser simplesmente descartadas, sem que ofereçam uma perda significativa de informações para o vídeo final.

Um processo amplamente utilizado para compressão de vídeo baseando-se na redundância temporal e espacial, é a Estimação de Movimentos (EM). A EM é responsável por encontrar uma correlação entre quadros, mapeando a redundância temporal entre os quadros vizinhos de uma cena (Porto, 2012). Após a eliminação do excesso de quadros da sequência da imagem, este processo visa processar e codificar apenas os macroblocos que se alteram entre um frame e outro.

### 2.3 REDUNDÂNCIA EM ENTROPIA

Enquanto a estimação de movimentos é responsável pela redução de redundâncias de *pixels* e quadros da imagem, a codificação em entropia é a responsável pela compressão a partir da redundância dos dados codificados. O objetivo principal é representar os dados de maior ocorrência na sequência pela menor representação binária possível (DUMA; FONSECA; POHL, 2010).

Esta redundância não tem uma relação direta com a informação das imagens a serem codificadas, mas sim com o tratamento dos dados obtidos através do processo de codificação da mesma. De acordo com o explicado por Porto (2012), a quantidade de informações geradas em um símbolo para transmissão é diretamente proporcional a probabilidade de ocorrência daquele símbolo. Com isto, símbolos que venham a aparecer muitas vezes deverão ser transmitidos com menos bits.

De maneira simplificada, pode-se adotar o método descrito por Gonzalez, Woods (2003), onde é descrito o processo de tratamento de redundância em entropia de uma imagem em escala de cinza da seguinte forma:

Entenda-se  $r_k$  como uma variável aleatória que atua no intervalo  $[0,1]$ , representando os níveis de cinza da imagem. Cada um destes níveis ocorre com uma probabilidade, que pode ser tratada como  $p_r(r_k)$ . Com isto, obtém-se a seguinte fórmula:

$$p_r(r_k) = \frac{n_k}{n} \quad k = 0, 1, 2, \dots, L - 1$$

Nesta fórmula que  $L$  é o número de níveis de cinza,  $n_k$  é o número de vezes que o  $k$ -ésimo nível de cinza aparece na imagem e  $n$  é o número total de pixels da imagem.

Através da estimativa de probabilidade de aparição de cada nível de cinza na imagem, quanto maior for a probabilidade, menor será a quantidade de bits utilizadas para representar aquele nível de cinza no processo de codificação da imagem.

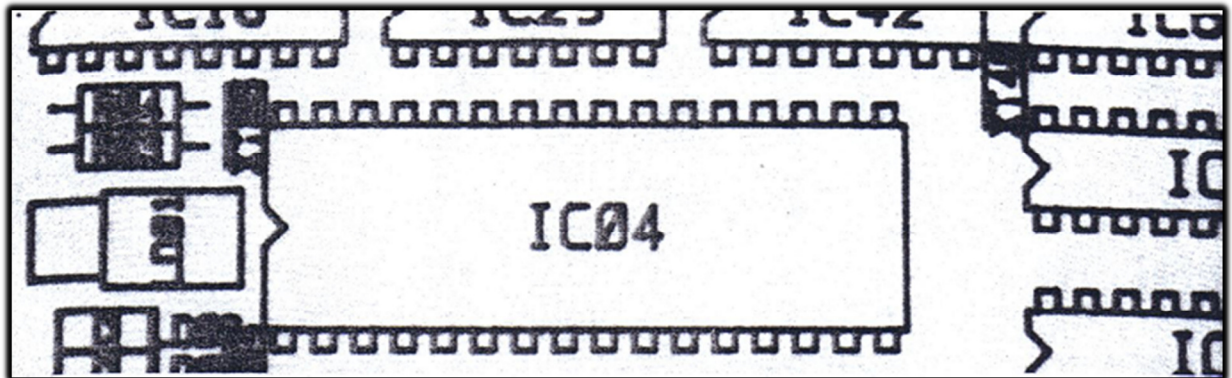
## 2.4 REDUNDÂNCIA PSICO-VISUAL

Como descrito anteriormente, o espaço de cores utilizado é o YCbCr, pois permite que seja feito um tratamento diferenciado entre os canais de luminância e

chrominância. A representação digital de um vídeo está associada à interpretação das cores pelo sistema visual humano. O sistema visual humano é mais sensível a informações de luminância do que a informações de crominância (VARGAS, 2007).

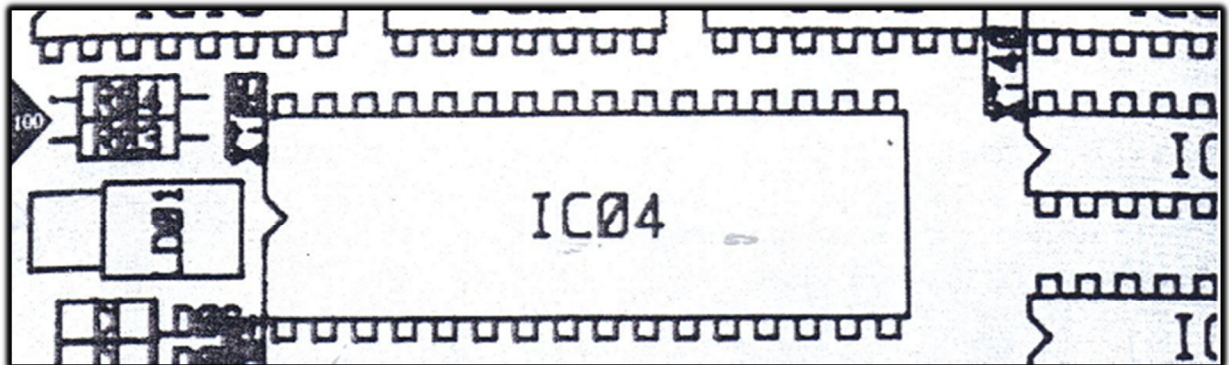
“A redundância psicovisual é fundamentalmente diferente das redundâncias anteriormente discutidas. Ao contrário da codificação e da redundância interpixels, a redundância psicovisual é associada com a informação visual quantificável ou real” (GONZALEZ; WOODS, 2003, p. 224). Considerando as limitações apresentadas pela visão humana, após a codificação dos sinais de luminância e crominância dos macroblocos apresentados nas imagens, pequenas variações dos mesmos podem ser descartadas sem causar perdas de conteúdo relevantes para o olho humano, porém que podem trazer uma significativa redução ao tamanho final do arquivo. Isto pode ser observado logo a seguir, onde a Figura 3 apresenta uma imagem com todos os níveis de cores presentes e a Figura 4 apresenta a mesma imagem porém com a redução dos níveis de cores, descartando assim informações de pouca relevância para o resultado final.

Figura 3 – Imagem monocromática com 256 níveis de cinza.



Fonte: Gonzalez e Woods (2003, p. 225).

Figura 4 – Imagem monocromática após redução para 16 níveis de cinza.



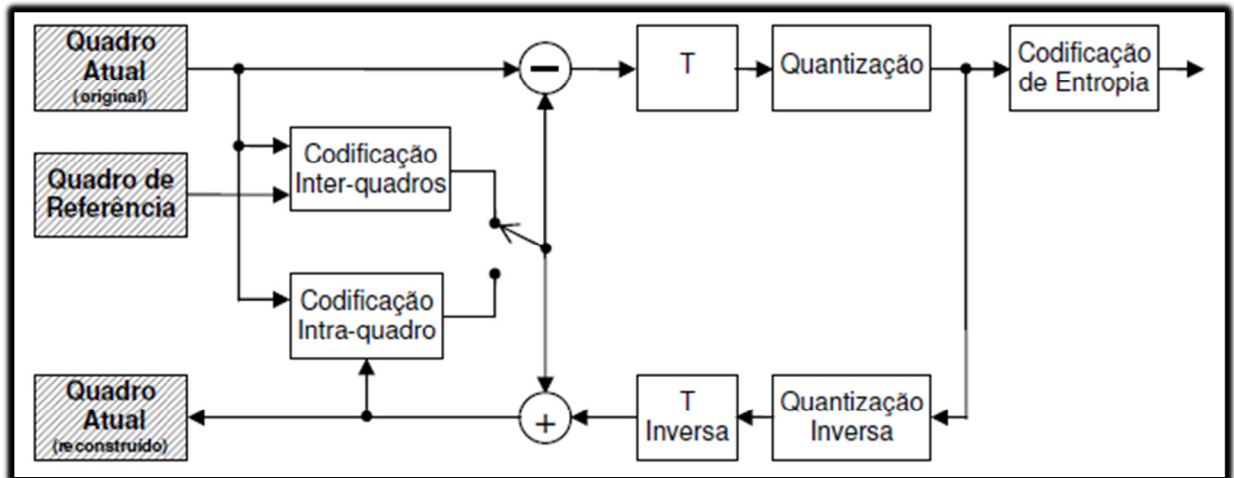
Fonte: Gonzalez e Woods (2003, p. 225).

## 2.5 CODIFICADOR DE VÍDEO

O codificador é o algoritmo que trabalha com os quadros do vídeo retirando as redundâncias de acordo com o citado anteriormente. Neste primeiro momento será mostrado um modelo resumido do codificador de vídeo, não englobando todo o escopo de tecnologia presente no padrão H.264/AVC que é o objeto de estudo deste trabalho.

Para isto, será seguido o exemplo de Agostini, 2007, em que divide os quadros a serem processados em dois tipos. Os quadros do tipo I são utilizados para sincronizar o sinal do vídeo, e só podem ser comprimidos aproveitando-se a redundância espacial (compressão intra-quadro), sem que as informações presentes nos quadros vizinhos influenciem no processo de compressão do mesmo. Isto se dá porque outros quadros serão construídos a partir deste, e, devido a isto, o máximo de informações devem ser preservadas. Os quadros do tipo P são os quadros resultantes do processo de codificação a partir dos quadros antecedentes. Para o processamento de quadros do tipo P para P, são utilizados além dos algoritmos de compressão intra-quadros, mais técnicas de eliminação de redundância temporal, realizando assim a codificação inter-quadros. Estas técnicas são divididas normalmente em duas etapas chamadas estimação de movimento e compensação de movimento (AGOSTINI, 2007). Um modelo mais detalhado, que define este modelo de codificador de vídeo, pode ser observado na Figura 5, cujos blocos serão explicados mais detalhadamente no Capítulo 4 desta obra.

Figura 5 – Diagrama de quadros de um codificador de vídeo.



Fonte: Agostini (2007, p. 34).

No capítulo a seguir será descrito o funcionamento do padrão H.264, cujo software de referência implementa o codificador seguindo o mesmo modelo apresentado na Figura 5.

### 3 O Padrão H.264/AVC

O padrão H.264/AVC de codificação de vídeo trouxe melhores resultados na compressão de vídeo comparado com os padrões anteriores, como MPEG-2, H.264 ou MPEG-4 Parte 2 Visual (RAVI, 2009). Foi desenvolvido pela JVT (Joint Vídeo Team), que consiste em um grupo de pesquisa formado por especialistas da VCEG e da MPEG. Este padrão traz uma significativa melhora na codificação, especificações com sintaxe simples e uma perfeita integração de codificação de vídeo com os protocolos atuais e arquiteturas multiplex. Assim, o H.264/AVC pode suportar diversas aplicações como transmissão de vídeo, *streaming* de vídeo, videoconferência através de redes fixas e sem fio e através de protocolos de transporte. (KOWN; TAMHANKAR; RAO, 2005).

A Tabela 1 traz uma relação de algumas características presentes no padrão H.264/AVC em comparação aos principais padrões de compressão de vídeos utilizados até então.

Tabela 1 – Comparação entre as diferentes características dos padrões H.264/AVC, MPEG-4 Parte 2 e MPEG-2.

	H.264/AVC	MPEG-4 Parte 2	MPEG-2
Tamanho de Macrobloco	16x16	16x16	16x16 (modo quadro) 16x8 (modo campo)
Tamanho de bloco	16x16, 8x16, 16x8, 8x8, 4x8, 8x4, 4x4	16x16, 16x8, 8x8	8x8
Predição Intra-quadro	Domínio Espacial	Domínio de Frequências	Não
Transformada	DCT Inteira 8x8, 4x4, Hadamard 4x4, 4x2	DCT 8x8/ Wavelet	DCT 8x8
Codificação de Entropia	VLC, CAVALC, CABAC	VLC	VLC
Precisão	1/4 pixel	1/4 pixel	1/2 pixel
Quadros de referência	Múltiplos quadros	Um quadro	Um quadro
Modo de predição bidirecional	Para frente/Para trás Para frente/Para frente Para trás/Para trás	Para frente/Para trás	Para frente/Para trás
Predição Ponderada	Sim	Não	Não
Filtro de Blocagem	Sim	Não	Não
Tipos de Quadros	I, P, B, SI, SP	I, P, B	I, P, B
Perfis	7 perfis	8 perfis	5 perfis
Taxa de transmissão	64 Kbps a 150 Mbps	64 Kbps a 2 Mbps	2 Mbps a 15 Mbps
Complexidade do Codificador	Alta	Média	Média

Fonte: Agostini (2007, p. 43).

Agostini (2007), também cita a tabela de comparação de bits entre o padrão H.264/AVC e os padrões MPEG-4 Parte 2, MPEG-2 e H.263 extraída da pesquisa de Wiegand (2003), e que pode ser observada na Tabela 2.

Tabela 2 – Comparativo da taxa de *bits* em relação ao padrão H.264/AVC.

Aplicação	MPEG-4 Parte 2 (%)	H.263 (%)	MPEG-2 (%)
Streaming	37,4	47,6	63,6
Vídeo Conferência	29,4	40,6	-
Entretenimento/Qualidade	-	-	45

Fonte: Wiegand (2003).

### 3.1 PERFIS E NÍVEIS

Os parágrafos a seguir descrevem os níveis em que o padrão H.264/AVC se divide com base na tese elaborada por Agostini, 2007.

Inicialmente, o padrão H.264/AVC foi dividido em três diferentes perfis: *Baseline*, *Main* e *Extended*, sendo cada um focado em uma aplicabilidade diferente.

O perfil *Baseline* tem suporte à codificação intra-quadros e inter-quadros utilizando *slices* do tipo I e P, e possui suporte também à codificação de entropia com códigos de comprimento de palavra variável adaptativas ao contexto (CAVLC). Este perfil é direcionado a aplicações de vídeo-telefonia, videoconferência e vídeo sem fio.

O perfil *Main*, além das tecnologias presentes no *Baseline*, possui ainda suporte a vídeo entrelaçado, codificação inter-quadros utilizando *slices* do tipo B e utilizando predição ponderada além de suportar também codificação aritmética adaptativa ao contexto (CABAC). Este perfil é voltado à utilização na transmissão de televisão e no armazenamento de vídeo.

O perfil *Extended*, por sua vez, não possui suporte para codificação de vídeo entrelaçado nem o CABAC, porém possui métodos para habilitar uma troca eficiente dentre *bitstreams* codificados através de *slices* de tipo SP e SI, e melhora a resiliência a erros utilizando o particionamento de dados.

Além das características descritas acima, estes perfis possuem também alguns pontos em comum. Em todos os três são utilizados 8 bits por amostra e a relação entre os elementos de cores é fixa, sendo definida em 4:2:0, ou seja, são formados por uma matriz 16x16 amostras de luminância e duas matrizes 8x8 de crominância, sendo uma para Cb e outra para Cr.

Estes três perfis iniciais, porém, não possuem suporte a vídeos de qualidade muito alta. Para resolver este problema, foi dada continuidade ao projeto JVT (responsável pela criação do modelo inicial do padrão H.264/AVC), o que resultou na criação de novas extensões para alcance de fidelidade que foram chamadas de *fidelity range extensions* – FRExt. Isto gerou um novo grupo com quatro novos perfis denominado perfis *High*.

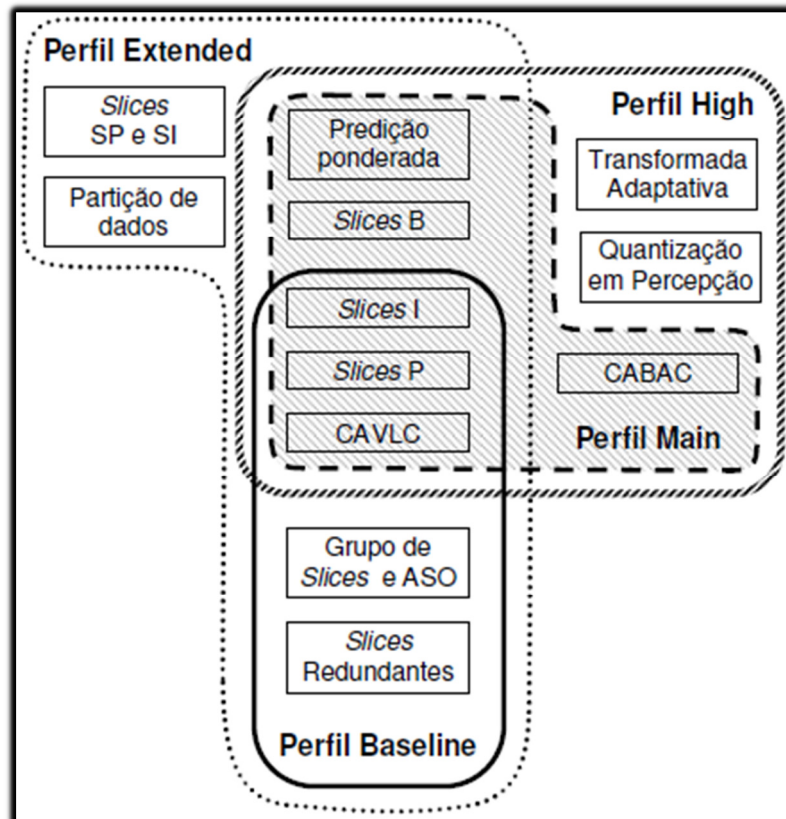
Nestes perfis *High* foram adicionados suporte a tamanho de bloco adaptativo para a transformada (4x4 ou 8x8), suporte a matrizes de quantização baseadas em percepção e suporte a representação e codificação sem perda de regiões específicas do conteúdo do vídeo.

Este novo grupo de perfis divide-se em:

- a) Perfil *High*, que inclui vídeos com 8 bits de amostra e com relação de cor 4:2:0;
- b) Perfil *High* 10, que inclui vídeos com até 10 bits de amostra e relação de cor 4:2:0;
- c) Perfil High 4:2:2, que suporta relação de cores 4:2:2 e vídeos com 10 bits de amostra.
- d) Perfil High 4:4:4, que apresenta suporte a relação de cores 4:4:4, e, devido a isto, possibilita a relação e codificação sem perdas de determinadas regiões do vídeo, e ainda possui suporte a vídeo com até 12 bits de amostra.

A Figura 6 apresenta uma relação resumida entre os perfis do padrão H.264/AVC descritos anteriormente.

Figura 6 – Perfis de codificação do padrão H.264/AVC.



Fonte: Agostini (2007, p. 41).

### 3.2 FORMATO DOS DADOS CODIFICADOS

Uma importante inovação trazida pelo H.264/AVC é a distinção feita entre a camada de codificação de vídeo (VCL) e a camada de abstração de rede (NAL).

Na VCL o vídeo é codificado e é gerada uma sequência de *bits* que é enviada para a unidade NAL, onde são repassadas para a transmissão ou para o armazenamento do mesmo (SILVA, 2007). Agostini (2007) descreve uma sequência de vídeo como sendo uma sequência de unidades NAL que podem ser transmitidas de algum modo ou ainda armazenadas em um arquivo.

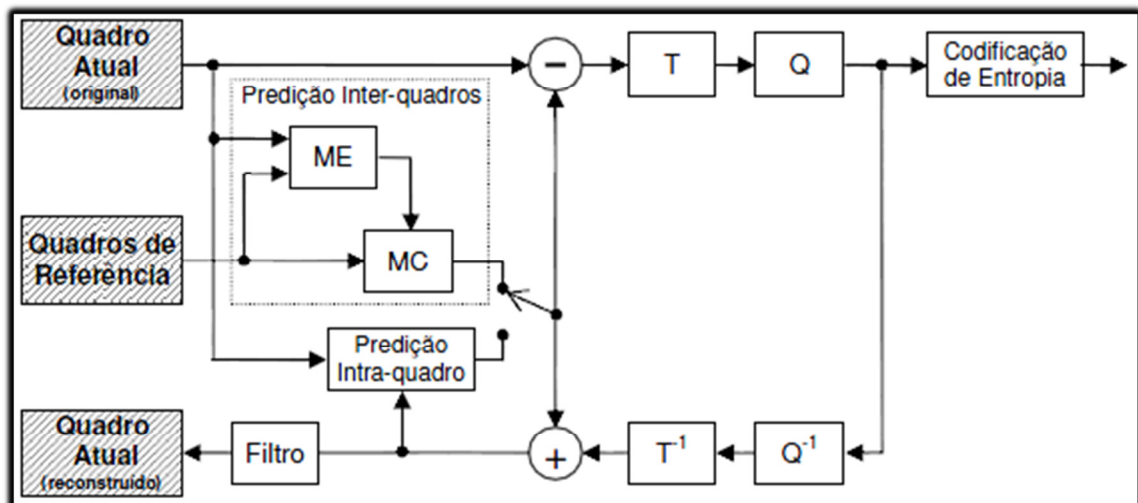
#### 4 Núcleo do Padrão H.264/AVC

Neste Capítulo será descrito o funcionamento interno do padrão H.264/AVC, com base nos estudos efetuados por Agostini (2007), sendo estes, complementados por relatos de outros autores.

Na Figura 7 tem-se o diagrama de blocos com os módulos presentes no codificador e decodificador. É possível notar que este diagrama é muito similar ao apresentado no Capítulo 2 desta pesquisa, porém traz algumas particularidades do padrão H.264/AVC.

“Uma característica interessante do padrão H.264/AVC é que apenas o decodificador é normatizado. Então vários graus de liberdade podem ser explorados na implementação do codificador” (AGOSTINI, 2007, p. 47). Ainda segundo Agostini (2007), como o codificador não é normatizado, torna-se livre a escolha dos algoritmos a serem utilizados no mesmo, desde que no final seja gerado um *bitstream* compatível com padrão entendido pelo decodificador.

Figura 7 – Diagrama de quadros do Padrão H.264/AVC.



Fonte: Agostini (2007, p. 45).

O processo de codificação se inicia pelo módulo de Predição Inter-quadros, onde estão presentes os blocos de estimação de movimento (ME – Motion Estimation) e compensação de movimento (MC – Motion Compensation). Nestes dois blocos estão presentes as maiores inovações e os maiores ganhos obtidos pelo padrão H.264/AVC em relação aos demais padrões comumente utilizados. Porém,

em virtude disso, estão também contidos os algoritmos de maior complexidade de toda a estrutura do codificador e decodificador. Mais adiante será explicado com maiores informações o funcionamento de cada um destes blocos.

Depois de feito os tratamentos de EM e MC, é feita a subtração dos valores do bloco original com os valores resultantes do processo de codificação. A informação resultante desta operação é chamada de resíduo. “O resíduo é a diferença dos elementos de luminância entre o quadro de referência e o quadro de vídeo atual (que está sendo codificado) [...]” (SILVA, 2007, p. 13).

A transformada (bloco T do diagrama de quadros apresentado na Figura 2) recebe o resíduo e é responsável pela eliminação das informações que não são relevantes para o olho humano, ou seja, retirar as informações que o olho humano não é capaz de distinguir das imagens. Neste bloco são reduzidas as redundâncias espaciais que permaneçam após o processo de codificação inter-quadros. Além disso, a transformada também é responsável pela conversão dos valores dos resíduos que estão no domínio espacial para o domínio de frequência, que podem obter resultados mais satisfatórios com os algoritmos de codificação de entropia, presentes do módulo de quantização.

No padrão H.264/AVC foram introduzidas algumas inovações para o módulo da transformada. Desde os perfis *Baseline*, *Main* e *Extended*, as dimensões da transformada foram definidas em 4x4. Nos perfis High, foi adicionado a capacidade de adaptação entre dimensões 4x4 ou 8x8. Outra inovação, é que uma segunda transformada é aplicada sobre os elementos resultantes da DCT para todos os blocos de crominância e para os macroblocos em que é feita a predição intraquadros 16x16. Ainda para a transformada, foi adicionada uma implementação em hardware de ponto fixo para evitar erros de casamento entre o codificador e o decodificador.

Como já mencionado anteriormente, para codificação de entropia o H.264/AVC foram adicionadas duas ferramentas que aumentam bastante a eficiência de codificação, que são os métodos CAVLC e CABAC. Para a codificação de entropia, a principal inovação introduzida na codificação aritmética é o uso de codificação adaptativa baseada em contextos. Neste método, o modo como são codificados os elementos sintáticos depende do elemento a ser codificado, da fase

em que se encontra o algoritmo de codificação e dos elementos sintáticos que já foram codificados (SILVA, 2007).

#### 4.1 MÓDULO DE ESTIMAÇÃO DE MOVIMENTO

O módulo de ME está somente no codificador e, apesar de ser o que apresenta maior complexidade e custo computacional, é o que apresenta os maiores ganhos de desempenho no padrão H.264/AVC com relação aos demais (RICHARDSON, 2003).

Na estimação de movimento, o codificador deve ser capaz de localizar nos quadros de referência ou no quadro atual, o bloco mais semelhante ao bloco que está sendo atualmente codificado e armazenar sua posição do quadro de referência deve ser armazenada em um vetor, chamado vetor de movimento, que deve ser inserido junto com a codificação do macrobloco atual. Assim, o bloco atual é representado pela diferença entre seus valores e os do bloco utilizado como referência. Esta diferença, geralmente, é menor que o valor original do bloco, trazendo assim uma redução na quantidade de bits necessários para a codificação do bloco (DARONCO, 2009).

“Para a realização da estimação de movimento é considerado apenas o componente de luminância do macrobloco” (AGOSTINI, 2007, p. 47).

Ainda analisando o descrito por Agostini (2007), a principal inovação da ME no H.264/AVC é a capacidade de utilizar blocos de vários tamanhos para a estimação de movimentos. De acordo com Richardson (2003), cada macrobloco pode ser dividido em sub-macroblocos de tamanho 16x16, 8x16, 16x8, e 8x8. Sendo que sub-macroblocos 8x8 ainda podem ser novamente subdivididos em 8x8, 8x4, 4x8 ou 4x4.

“Este método de particionar macroblocos em sub-blocos de tamanho variável é conhecido como compensação de movimento estruturada em árvore” (AGOSTINI, 2007, p 48).

Para cada sub-macrobloco criado, deve ser gerado também um novo vetor de movimentos, que estará presente no *bitstream* gerado na codificação do bloco atual juntamente com o tipo de partição escolhida. Deste modo, quanto mais

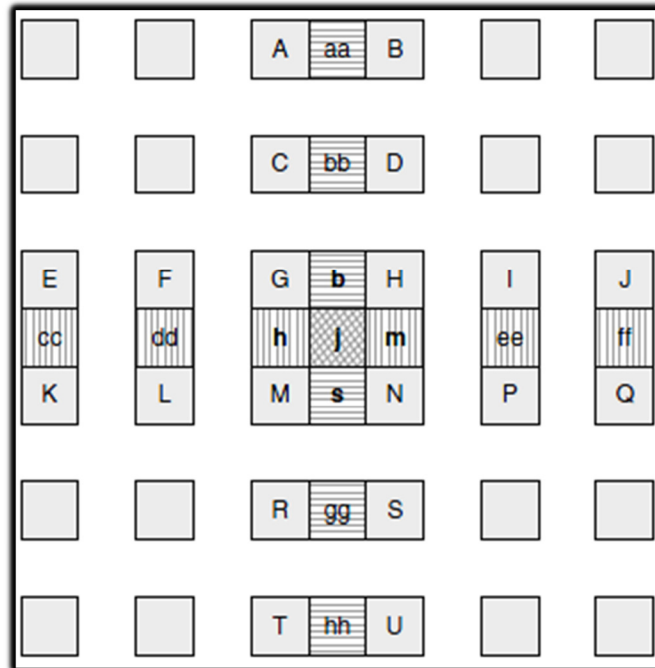
sub-macroblocos forem criados, maior será a quantidade de bits necessários para efetuar a codificação, o que resulta em uma menor taxa de compressão dos dados. Por outro lado, a subdivisão de macroblocos grandes (16x16, 16x8, 8x16) pode trazer ganhos consideráveis em áreas com muitos detalhes através da compensação de movimentos. Devido a isto, o codificador deve ser capaz de efetuar uma avaliação entre a necessidade de subdividir ou não um macrobloco grande. Este é o ponto mais crítico da implementação dos algoritmos de ME, uma vez que é feita a análise de todos os tamanhos de blocos e no final somente um é o escolhido (AGOSTINI, 2007; RICHARDSON, 2003).

Outra característica da estimação de movimentos presente no H.264/AVC e descrita por Richardson (2003) e Agostini (2007), é a precisão de  $\frac{1}{4}$  de pixel para vetores de movimento, onde o vetor de movimento é obtido através da interpolação entre o bloco atual e o bloco de referência. Isto é importante porque geralmente os movimentos que ocorrem entre um quadro e outro não estão exatamente em posições inteiras de pixel, e com isto, se não forem suportados valores fracionados, torna-se bem mais difícil encontrar casamentos bons para a ME.

Agostini (2007), ainda frisa que para o padrão H.264/AVC, a estimação de movimentos utilizando  $\frac{1}{4}$  de pixel é obrigatória, porém, na prática,  $\frac{1}{4}$  de pixel é utilizado para os componentes de luminância, sendo que para os de crominância é utilizada uma precisão de  $\frac{1}{8}$  de pixel.

A interpolação para  $\frac{1}{4}$  de pixel é executada em dois passos. Primeiro, é executada a interpolação para  $\frac{1}{2}$  pixel utilizando um filtro FIR (*Finite Impulse Response*) de seis *taps* com pesos (1/32, -5/32, 5/8, 5/8, -5/32, 1/32) (AGOSTINI, 2007), conforme apresentado na Figura 8.

Figura 8 – Interpolação para  $\frac{1}{2}$  de pixel.



Fonte: Richardson (2003, p. 173).

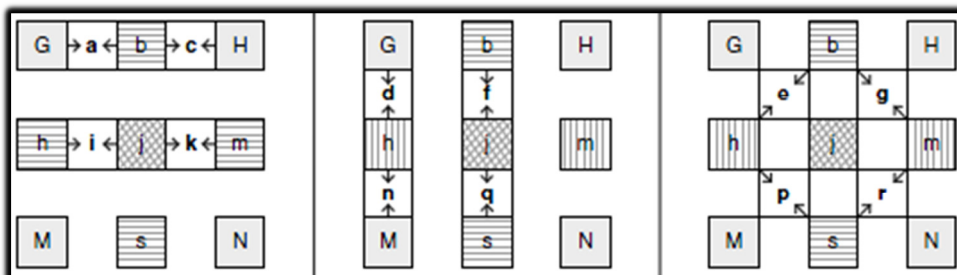
Exemplificando,  $b$ , na Figura 8, é calculado a partir da seguinte fórmula:

$$b = (E - 5F + 20G + 20H - 5I + J)/32$$

De modo similar,  $h$  é interpolado pela filtragem de  $A$ ,  $C$ ,  $G$ ,  $M$ ,  $R$  e  $T$ . A mesma regra é válida para calcular o valor de  $j$ , realizando assim a interpolação entre  $cc$ ,  $dd$ ,  $h$ ,  $m$ ,  $ee$  e  $ff$ , ou então entre  $aa$ ,  $bb$ ,  $b$ ,  $s$ ,  $gg$  e  $hh$ , uma vez que calculando horizontal ou verticalmente, o resultado obtido deverá ser o mesmo (RICHARDSON, 2003).

O segundo passo é a interpolação para  $\frac{1}{4}$  de pixel, seguindo o exposto na Figura 9.

Figura 9 – Interpolação para  $\frac{1}{4}$  de pixel.



Fonte: Richardson (2003, p. 174).

A interpolação para  $\frac{1}{4}$  de pixel é obtida através da média entre os valores inteiros e os valores resultantes da interpolação para  $\frac{1}{2}$  (AGOSTINI, 2007). Isto pode ser exemplificado, através das fórmulas para cálculo dos valores de  $a$ ,  $d$  e  $p$ , sendo estas as seguintes:

$$a = \frac{G + b}{2}; \quad d = \frac{G + h}{2}; \quad p = \frac{h + s}{2}$$

Outro ponto importante a ser ressaltado sobre o padrão H.264/AVC, é a capacidade de utilizar múltiplos quadros para a estimação de movimentos, não se restringindo somente aos *slices* de tipo I e P imediatamente adjacentes ao quadro atual. Com este padrão é possível realizar a estimação de movimentos baseados em múltiplos quadros para frente ou para trás do quadro atual (RICHARDSON, 2003).

É possível utilizar quadros que ainda não foram exibidos no vídeo para referência da estimação de movimentos. Isto se dá pelo fato de que no padrão H.264/AVC, o codificador define uma ordem de codificação dos quadros diferente da ordem de exibição dos mesmos. Com isto, um quadro que ainda não foi exibido, pode já ter sido codificado, o que torna possível que este quadro seja utilizado como referência para a estimação de movimentos (AGOSTINI, 2007).

#### 4.2 MÓDULO DE COMPENSAÇÃO DE MOVIMENTO

“O Compensador de Movimento (MC) no codificador H.264/AVC é uma etapa precedida pela etapa de Estimação de Movimento (ME)” (SILVA, 2007, p. 23 e 24). Ainda seguindo o exposto por Silva (2007), a compensação de movimento consiste em um processo cujo objetivo é compensar o deslocamento de objetos movidos de um quadro para outro. A função da MC no codificador é montar o quadro predito, ou seja, buscar os quadros de melhor casamento para os blocos oriundos da ME através dos vetores de movimento. Depois de gerado o quadro predito, este é subtraído do quadro atual para obter o resíduo que será enviado ao módulo da transformada.

Por ser precedida pela ME, no codificador a MC pode ser simplificada. Porém, no decodificador a MC deve ser completa, estando “apta a operar sobre

todas as funcionalidades do padrão” (AGOSTINI, 2007, p. 53), ou seja, deve ser capaz de trabalhar com todos os valores que possam ser gerados para o *bitstream* realizando a reconstrução correta dos blocos.

### 4.3 O PADRÃO H.264/SVC

O padrão H.264/AVC possui uma extensão desenvolvida que visa o suporte a escalabilidade denominado H.264/SVC – *Scalable Video Coding* (Wiegand et al., 2007). De acordo com os estudos apresentados por Daronco (2009), a primeira apresentação do H.264/SVC foi feita em setembro de 2007. Este novo padrão apresentava a solução para os problemas enfrentados até então com a codificação de vídeo escalável.

Soler (2006) descreve a escalabilidade como sendo a capacidade de através do decodificador se extrair de um *bitstream* subconjuntos de *bits* com variações de alguma característica da imagem ou vídeo, como por exemplo, resolução espacial ou qualidade de imagem.

Ainda em sequência ao relato de Soler (2006), seguindo a definição de escalabilidade como sendo a capacidade de se extrair diferentes características de um vídeo ou imagem a partir de um *bitstream*, a soma de um conjunto de *bitstreams* diferentes poderia representar a característica de escalabilidade. Porém, para que um *bitstream* possa ser caracterizado como escalável, devido à necessidade de eficiência, para a obtenção de escalabilidade um *bitstream* base deve ser utilizado, recebendo *bitstreams* complementares contendo somente as alterações necessárias, como resoluções ou índices de qualidade diferentes.

Com isto, podem-se elencar como principais melhorias deste padrão, o uso de técnicas de predição de movimentos mais avançadas, aumentando assim a eficiência da codificação; redução na complexidade de codificação e a utilização do próprio H.264 para codificação, que resulta em um considerável ganho de desempenho na codificação de vídeo.

Daronco (2009) frisa também que, apesar de o termo “escalabilidade” possuir uma abrangência bem grande quanto a seu significado, para o padrão H.264 é adotada a ideia de adaptação do fluxo de vídeo as necessidades e capacidades do

usuário, de modo com que seja possível a remoção de partes do fluxo do vídeo para que as necessidades sejam mais bem atendidas.

Isto significa que O H.264/SVC provê suporte a múltiplas resoluções de exibição com um único *bitstream*. Adicionalmente, o H.264/SVC também suporta escalabilidade temporal, que está relacionada com a taxa de quadros, ou seja, alguns quadros podem ser descartados; e a escalabilidade qualitativa, que é o enriquecimento na fidelidade dos quadros mantendo-se a resolução entre as camadas. O H.264/SVC também permite a combinação dessas técnicas (SEGALL; SULLIVAN, 2007).

Wien, Schwarz e Oelbaum (2007), em sua análise de performance do H.264/SVC, puderam concluir que este padrão não apresentou perdas de desempenho relevantes com relação ao padrão H.264/AVC, agregando as características presentes em métodos de codificação de vídeo escalável com significativa redução de complexidade.

## **5 Trabalhos Correlatos**

### **5.1 AVALIAÇÃO DE DESEMPENHO DA ESCALABILIDADE ESPACIAL NA RECOMENDAÇÃO H.264/SVC PARA CODIFICAÇÃO DE VÍDEO**

Desenvolvido por Márlon Henrique de Souza para a obtenção do grau de Bacharel em Ciências da Computação pela Universidade do Extremo Sul Catarinense, este trabalho objetivou o estudo da escalabilidade espacial aplicada ao padrão H.264, através de uma análise qualitativa e quantitativa, utilizando como software de referência o JSVM, de seu desempenho e as vantagens possibilitadas a partir deste para o uso em dispositivos de baixa resolução.

### **5.2 PERFORMANCE ANALYSIS OF SVC**

Este relato foi desenvolvido em 2007, por Mathias Wien (membro da IEEE), Heiko Schwarz e Tobias Oelbaum, logo após a divulgação do primeiro modelo do H.264/SVC. A pesquisa foi realizada com o intuito de fazer uma análise de performance deste padrão H.264/SVC com relação ao H.264/AVC e aos demais métodos de codificação de vídeo escalável, como MPEG-4 e ASP, além de abordar alguns aspectos como problemas que podem ser encontrados e métodos de tratamento para os mesmos.

### **5.3 COMPARISON OF STEREO VIDEO CODING SUPORT IN MPEG-4 MAC, H.264/AVC AND H.264/SVC**

Desenvolvido por C. T. E. R. Hewage, H. A. Karim, S. Worrall, S. Dogan e A. M. Kondo, este artigo tem como foco a codificação de vídeos voltados a filmagens 3D. Os pesquisadores envolvidos no desenvolvimento deste efetuaram uma análise comparativa de resultados para avaliar quais as vantagens e desvantagens entre os padrões MPEG-4, H.264/AVC e H.264/SVC na codificação de imagens capturadas em 3D, onde obtiveram resultados que comprovaram que para este tipo de tecnologia, a codificação escalável apresenta um considerável ganho de performance e qualidade.

#### 5.4 UM ESTUDO SOBRE O PADRÃO H.264/AVC DE COMPRESSÃO DE VÍDEO

Esta dissertação para obtenção do Grau de Pós-Graduação em Informática pela Universidade Católica de Pelotas foi desenvolvida por André Marcelo Coelho da Silva, para a obtenção do grau de pós-graduação em informática no ano de 2007 e teve como principal foco um estudo mais aprofundado sobre padrão de codificação de vídeo H.264/AVC. Antes de aprofundar os estudos sobre o padrão, o autor apresentou conceitos básicos sobre codificação e compressão de vídeo, que servem como base para o entendimento do funcionamento dos módulos que compõem o H.264/AVC, e que foram descritos posteriormente, sendo estes o foco principal de estudo do trabalho.

## 6 Comparação dos Codificadores

Neste Capítulo serão descritos os métodos de extração de dados para a avaliação de cada codificador, as sequências de vídeo utilizadas nos testes bem como os resultados obtidos com a codificação de cada sequência de vídeo.

### 6.1 METODOLOGIA

A extração de dados para a análise a ser efetuada nesta pesquisa foi feita através dos *softwares* de referência para codificação de vídeo H.264 com e sem suporte a escalabilidade. Nas sessões a seguir serão detalhadas as configurações utilizadas para a extração dos dados.

#### 6.1.1 Joint Scalable Video Model (JSVM)

As informações apresentadas nesta sessão sobre a JSVM e suas configurações foram buscadas no manual de referência disponibilizado pelo JVT juntamente com o código fonte do software.

Atualmente na versão 9.19.14, liberada em 14 de junho de 2011, o JSVM foi projetado pelo *Joint Video Team (JVT)* da *ISO/IEC Moving Pictures Experts Group (MPEG)* e a *ITU-T Video Coding Experts Group (VCEG)* e é a referência para a codificação de vídeo escalável (SVC).

Foi desenvolvido em C++ e em sua última versão possui suporte a sistemas operacionais (SO) 32 bits e 64 bits. A construção do projeto e a amostragem de dados desta obra foram obtidas utilizando um SO de 64 bits.

O projeto JSVM possui vários executáveis para manipular os arquivos de vídeo, porém serão abordados para a análise dos dados somente três. São eles:

- a) H264AVCEncoderLibTestStatic: Este executável é o responsável por efetuar a codificação de um arquivo no formato YUV com ou sem suporte a escalabilidade.

- b) BitStreamExtractorStatic: Com este executável é possível efetuar a decodificação de um vídeo com menor fluxo de bits que o original, definindo qual camada será extraída.
- c) H264AVCDecoderLibTestStatic: Este é o decodificador, que recebe o arquivo no formato H.264 gerado pelo H264AVCEncoderLibTestStatic e efetua a decodificação. Funciona tanto para vídeos em formato escalável quanto não escalável.

#### 6.1.1.1 Codificação de Vídeo pela JSVM

Como já descrito anteriormente, para efetuar a codificação do sinal digital de um arquivo de vídeo, deve ser utilizado o executável H264AVCEncoderLibTestStatic, que receberá como parâmetro um arquivo de configuração contendo as informações necessárias para a geração dos resultados, o qual será descrito na sessão a seguir. Para realizar a chamada do executável, deve-se seguir o exemplo descrito na Figura 10.

Figura 10 – Comando de chamada do codificador.

```
H264AVCEncoderLibTestStatic.exe -pf <mcfg> [command line options]
```

Fonte: ITU-T & ISO/IEC JVT - Manual JSVM 9.19.14.

Conforme apresentado na Figura 10, o parâmetro “-pf” é obrigatório, pois é utilizado para a chamada do arquivo que contém as configurações a serem utilizadas pelo codificador. Como com a JSVM é possível codificar vídeos com e sem suporte a escalabilidade, para cada uma destas opções o arquivo de configurações deve seguir um padrão diferente. Os demais parâmetros são detalhados no manual da JSVM 9.19.14. A seguir serão apresentadas algumas informações sobre os principais campos do arquivo de configuração principal e dos arquivos de cada camada específica somente para codificação com suporte a escalabilidade (SVC).

##### 6.1.1.1.1 Arquivos de configurações para a JSVM

Este arquivo contém os parâmetros gerais que deverão ser adotados para todas as camadas de codificação do sinal de vídeo, além de conter também a chamada para os arquivos de cada camada específica (*layer*), onde são definidos os parâmetros específicos de cada camada. A Tabela 3 apresenta os principais parâmetros a serem considerados pela análise apresentada nesta obra.

Tabela 3 – Arquivo principal de configuração da JSVM.

<b>Parâmetro</b>	<b>Valor</b>	<b>Descrição</b>
AVCMode	0	SVC = 0; AVC = 1 (Parâmetro opcional, <i>default</i> = 0)
OutputFile	exemplo.264	Caminho do arquivo de saída no formato H.264
FrameRate	30.0	Frames por segundo
FramesToBeEncoded	30	Número de frames a serem codificados
GOPSize	1	Número de camadas em escala temporal
NumLayers	2	Número de camadas em escala espacial
LayerCfg	layer_0.cfg	Arquivo de configuração de cada camada
LayerCfg	layer_1.cfg	Arquivo de configuração de cada camada

Fonte: Do autor.

Para que seja efetuada a simulação espacial, devem ser definidos no mínimo dois níveis de camada de codificação (*layers*). Os detalhamentos dos principais parâmetros que serão considerados na análise estão descritos a seguir nas Tabelas 4 e 5.

Tabela 4 – Arquivo de configuração do *layer* 0.

<b>Parâmetro</b>	<b>Valor</b>	<b>Descrição</b>
SourceWidth	176	Largura da sequência de entrada
SourceHeight	144	Altura da sequência de entrada
InputFile	ex_qcif.yuv	Sequência de entrada específica da camada.
ReconFile	ex_qcif_layer0.yuv	Arquivo YUV reconstruído
QP	32.0	Parâmetro de quantização

Fonte: Do autor.

Tabela 5 – Arquivo de configuração do *layer* 1.

<b>Parâmetro</b>	<b>Valor</b>	<b>Descrição</b>
SourceWidth	352	Largura da sequência de entrada

SourceHeight	288	Altura da sequência de entrada
InputFile	ex_cif.yuv	Sequência de entrada específica da camada.
ReconFile	ex_cif_layer1.yuv	Arquivo YUV reconstruído
QP	32.0	Parâmetro de quantização

---

Fonte: Do autor.

### 6.1.1.1.2 Resultados gerados pela codificação

Durante o processo de codificação, a JSVM apresenta uma Tabela com dados estatísticos a respeito de cada quadro que está sendo codificado, como pode ser observado na Figura 11. Cada uma das colunas destacadas é descrita na Tabela 6. Ao final é apresentada a amostragem da média dos resultados obtidos para cada camada, conforme mostrado na Figura 12 e são gerados os arquivos reconstruídos de cada *layer* no formato YUV. Além disto, é gerado também um arquivo no formato \*.264 cujo nome será o definido pelo parâmetro “OutputFile” do arquivo de configuração utilizado pelo codificador.

Figura 11 – Apresentação de frames codificados em tempo real.

	C1		C2	C3	C4	C5	C6			C7
AU	0: I	T0	L0	Q0	QP 40	Y 28.1288	U 40.0329	V 41.0214	5600 bit	
	0: I	T0	L1	Q0	QP 51				472 bit	
AU	1: P	T0	L0	Q0	QP 40	Y 25.0604	U 39.6273	V 40.7215	472 bit	
	1: P	T0	L1	Q0	QP 51	Y 28.0311	U 40.0699	V 41.0006	368 bit	
AU	2: P	T0	L0	Q0	QP 40	Y 24.9806	U 39.7096	V 40.7333	288 bit	
	2: P	T0	L1	Q0	QP 51	Y 27.9201	U 40.0756	V 41.1214	296 bit	
AU	3: P	T0	L0	Q0	QP 40	Y 24.8737	U 39.7804	V 40.8413	496 bit	
	3: P	T0	L1	Q0	QP 51	Y 27.7999	U 40.1163	V 41.0817	280 bit	
AU	4: I	T0	L0	Q0	QP 40	Y 24.7980	U 39.8427	V 40.8043	288 bit	
	4: I	T0	L1	Q0	QP 51	Y 28.1385	U 40.5202	V 40.4745	480 bit	
AU	5: P	T0	L0	Q0	QP 40	Y 25.0112	U 40.1215	V 40.1716	312 bit	
	5: P	T0	L1	Q0	QP 51	Y 28.0586	U 40.5253	V 40.4126	320 bit	
AU	6: P	T0	L0	Q0	QP 40	Y 24.9399	U 40.1489	V 40.1673	408 bit	
	6: P	T0	L1	Q0	QP 51	Y 27.9066	U 40.4821	V 40.2875	296 bit	
AU	7: P	T0	L0	Q0	QP 40	Y 24.7923	U 40.0832	V 40.0013	304 bit	
	7: P	T0	L1	Q0	QP 51	Y 27.8325	U 40.4455	V 40.0280	552 bit	
AU	7: P	T0	L0	Q0	QP 40	Y 24.7239	U 40.0681	V 40.0155	280 bit	
	7: P	T0	L1	Q0	QP 51				288 bit	

Fonte: Do autor.

Tabela 6 – Descrição das colunas da Figura 11.

Coluna	Descrição
C1	Frame que está sendo codificado
C2	Camada em escala temporal
C3	Camada em escala espacial
C4	Camada em escala de qualidade
C5	Parâmetro de quantização estipulado para a camada
C6	Valores de PSNR
C7	Bit usados na codificação

Fonte: Do autor.

Figura 12 – Resultados ao término da codificação.

SUMMARY :						
	bitrate	Min-bitr	Y-PSNR	U-PSNR	U-PSNR	
176x144 @ 15.0000	681.8320	681.8320	38.1461	36.3571	36.8712	
352x288 @ 15.0000	687.6080	687.6080	11.5454	19.5036	19.7400	
Encoding speed: 166.333 ms/frame, Time:2495.000 ms, Frames: 15						

Fonte: Do autor.

### 6.1.1.2 Decodificação do Sinal Gerado Pelo Codificador

Após obtermos as sequências de *bitstream* o arquivo de formato \*.264 gerado pelo codificador deve ser enviado para o decodificador para que seja gerado um arquivo no formato YUV. A chamada deste executável recebe apenas 2 parâmetros, conforme descrito na Figura 13.

Figura 13 – Chamada do decodificador da JSVM.

```
H264AVCDecoderLibTestStatic <str> <rec> [-ec <ec>] [<maxPOCDiff>]
```

Fonte: Do autor.

O argumento <str> apresentado na Figura 13 corresponde ao arquivo de entrada, no formato \*.264, e o argumento <rec> corresponde ao arquivo que será gerado pelo decodificador, no formato YUV.

### 6.1.2 Codificação de vídeo com JM

Para a codificação de vídeo sem suporte a escalabilidade (AVC) foi utilizado o software de referência H.264/14496-10 AVC, e neste Capítulo serão apresentadas algumas características deste SW para que seja possível efetuar a codificação de um vídeo sem suporte a escalabilidade de acordo com o manual do mesmo.

A JM encontra-se atualmente na versão 18.5, liberada em março de 2011, foi desenvolvida em C++ e possui suporte a arquiteturas 32 e 64 bits. Esta pesquisa efetuará testes somente com arquitetura 64 bits.

Ao compilarmos o projeto da JM com o Microsoft Visual Studio 2012, foram gerados os executáveis necessários para a manipulação dos sinais de vídeo.

Dentre eles, serão descritos o `lencod.exe` e o `ldecod.exe`. O foco desta pesquisa será mantido o `lencod.exe`, que efetua a codificação e apresenta dados estatísticos que serão utilizados na análise dos resultados da codificação de vídeo.

#### 6.1.2.1 O Codificador JM

Do mesmo modo como funciona na JSVM, o codificador da JM também utiliza um arquivo de configurações para que possa efetuar a codificação do vídeo. Por padrão existem juntamente com o projeto alguns arquivos no formato `*.cfg` que servem como exemplo, porém podem ser utilizados arquivos diferentes, desde que sigam o mesmo padrão apresentado nos arquivos de exemplo. A chamada do codificador se dá de acordo com o apresentado na Figura 14.

Figura 14 – Chamada do codificador da JM.

```
lencod [-h] [-d defenc.cfg] {[-f curenc1.cfg]...[-f curencN.cfg]}
      {[-p EncParam1=EncValue1]...}
```

Fonte: Do autor.

Conforme apresentado na Figura 14, existem alguns parâmetros que podem ser passados para o encoder. A Tabela 7 apresenta uma breve descrição a respeito de cada um dos parâmetros.

Tabela 7 – Parâmetros do codificador da JM.

Parâmetro	Descrição
<b>-h</b>	Apresenta os parâmetros utilizados
<b>-d</b>	Utiliza <code>&lt;defenc.cfg&gt;</code> como arquivo de configuração. Caso não seja informado este parâmetro, o codificador vai buscar o arquivo "encoder.cfg" no diretório onde se encontra o executável.
<b>-f</b>	Lê o arquivo <code>&lt;curencM.cfg&gt;</code> para repor os parâmetros de codificador selecionados. Vários arquivos podem ser usados que definir parâmetros diferentes.

Fonte: Do autor.

Nesta pesquisa foi utilizado somente o parâmetro “-d”, onde se especifica o arquivo com as configurações principais para o codificador. Neste arquivo, foram mantidos quase todos os parâmetros padrões, já definidos no arquivo “encoder.cfg” disponível junto com a ferramenta, porém alguns foram revisados e estão descritos na Tabela 8.



Tabela 8 – Principais campos do arquivo de configuração.

Parâmetro	Valor	Descrição
InputFile	foreman_cif_352x288.yuv	Arquivo de entrada
InputHeaderLength	0	Tamanho do cabeçalho da sequência de entrada
StartFrame	0	Frame de início da codificação
FramesToBeEncoded	300	Quantidade de frames a codificar
FrameRate	30.0	Frames por segundo
SourceWidth	352	Largura da sequência de entrada
SourceHeight	288	Altura da sequência de entrada
OutputWidth	352	Largura da sequência de saída
OutputHeight	288	Largura da sequência de entrada
ReconFile	teste.yuv	Arquivo YUV reconstruído
OutputFile	test.264	Arquivo de <i>bitstream</i>

Fonte: Do autor.

Durante a codificação, o “lencod” apresenta na tela uma tabela com os resultados de qualidade de codificação de cada frame, conforme Figura 15. Depois de concluído é gerado o arquivo “stats.dat”, que contém informações mais detalhadas a respeito do processo de codificação. Este arquivo pode ser visualizado por qualquer editor de texto.

Figura 15 – Tabela de resultados em tempo real do codificador JM.

Frame	Bit/pic	QP	SnrY	SnrU	SnrV	Time<ms>	MET<ms>	Frn/Fld	Ref
00000<NUB>	320								
00000<IDR>	55424	28	36.681	35.957	36.083	434	0	FRM	3
00008< P >	21584	28	35.973	34.648	34.412	655	173	FRM	2
00004< B >	1488	30	35.482	32.621	31.441	1211	655	FRM	1
00002< B >	17584	31	33.727	36.686	37.473	2962	2177	FRM	1
00001< B >	25464	32	32.323	32.424	32.245	2884	2121	FRM	0
00003< B >	3576	32	41.217	38.539	36.392	2136	1646	FRM	0
00005< B >	3944	31	33.420	35.570	37.413	2494	1875	FRM	1
00005< B >	26128	32	33.051	32.444	32.277	3871	2481	FRM	0
00007< B >	3488	32	41.415	38.259	36.112	2388	1887	FRM	0
00016< P >	28424	28	36.267	35.888	35.118	1618	1132	FRM	2
00012< B >	928	30	35.282	33.157	31.038	1484	843	FRM	1
00018< B >	5216	31	33.348	36.594	37.226	2575	1953	FRM	1
00009< B >	26736	32	33.171	32.648	32.386	3895	2398	FRM	0
00011< B >	3476	32	41.294	38.847	36.105	2319	1826	FRM	0

Fonte: Do autor.

Para que fique mais fácil de serem compreendidas as informações apresentadas na Figura 15, as colunas apresentadas na imagem são descritas na Tabela 9.

Tabela 9 – Colunas dos resultados apresentados pelo codificador da JM.

Coluna	Descrição
Frame	A ordem de apresentação e o tipo do frame codificado
Bit/pic	Bits alocados para o frame atual
QP	Valor de quantização de frames
SnrY	Luma Y PSNR
SnrU	Chroma U PSNR
SnrV	Chroma V PSNR
Time	Tempo total de codificação do frame
MET	Tempo total de estimação de movimento do frame
Frm/Fld	Modo de codificação de imagem
Ref	Indicador se é um frame de referência

Fonte: Do autor.

Ao término da codificação, além do arquivo “stats.dat” é apresentado um resumo do resultado final, com o tempo total e uma média dos valores resultantes da codificação do vídeo, como é apresentado pela Figura 16.

Figura 16 – Média dos resultados gerados pelo codificador JM.

```

----- Average data all frames -----
Total encoding time for the seq. : 730.785 sec (0.41 fps)
Total ME time for sequence      : 550.275 sec

Y ( PSNR (dB), cSNR (dB), MSE )  : ( 36.113, 35.026, 28.44863 )
U ( PSNR (dB), cSNR (dB), MSE )  : ( 35.170, 34.604, 22.52410 )
V ( PSNR (dB), cSNR (dB), MSE )  : ( 34.051, 34.480, 23.61026 )

Total bits                       : 4203056 (I 55424, P 1072360, B 3155744)
320
Bit rate (kbit/s) @ 30.00 Hz     : 420.39
Bits to avoid Startcode Emulation : 2068
Bits for parameter sets         : 320
Bits for filler data            : 0

-----
Exit JM 18 (PRExt) encoder ver 18.5

```

Fonte: Do autor.

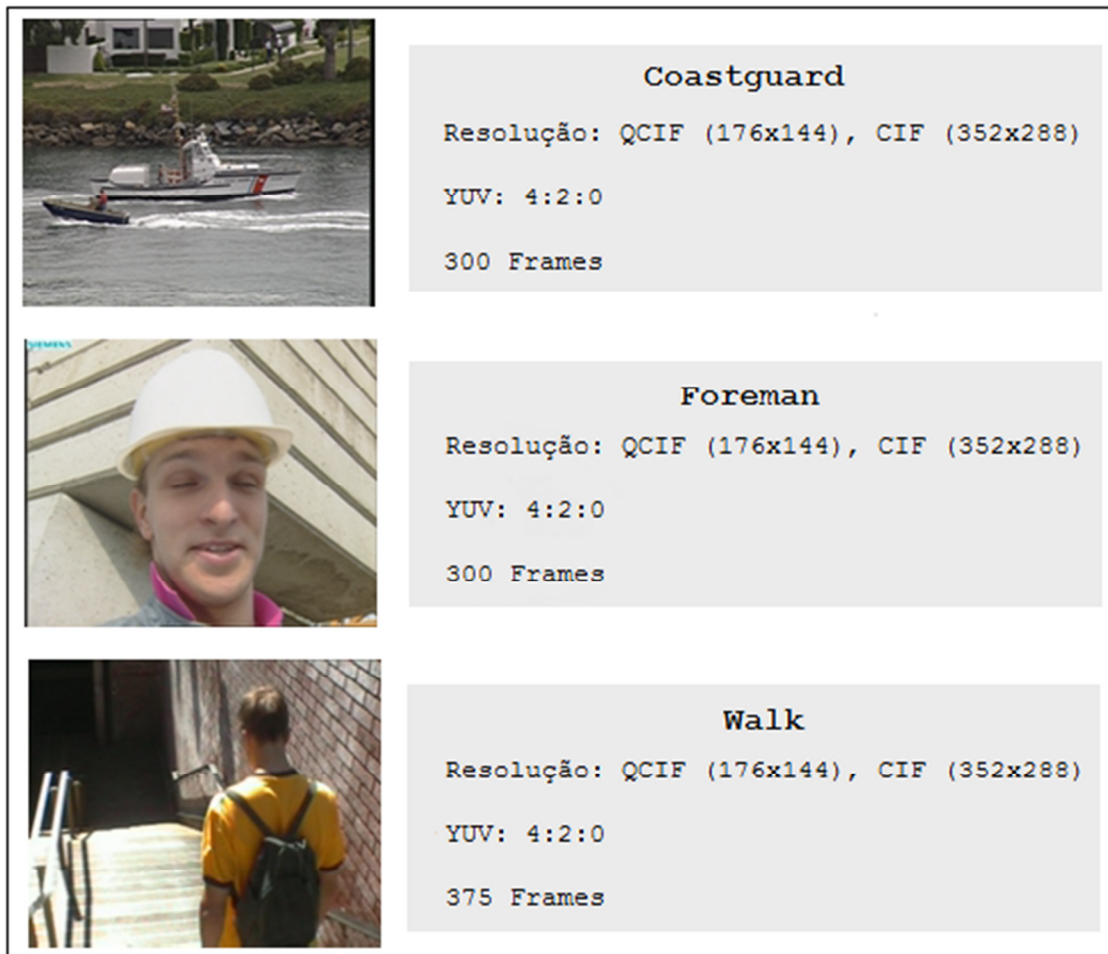
### 6.1.3 Sequências de vídeo utilizadas para testes

Os dados utilizados na comparação de resultados dos codificadores serão extraídos das sequências de vídeo *Coastguard*, *Foreman* e *Walk*, comumente utilizadas na análise de desempenho de codificação de vídeo. Para a análise dos resultados sem escalabilidade, foram utilizados vídeos no formato QCIF, e para a comparação de resultados utilizando o suporte a escalabilidade do SW JSVM foram utilizados vídeos no formato CIF. Optou-se por utilizar estas três sequências de vídeo por apresentarem um bom nível de abrangência nos testes, uma vez que cada uma apresenta um diferente nível de complexidade e de informações a serem processadas pelo codificador.

A sequência *Coastguard* consiste na filmagem de dois barcos navegando em um rio. No início do vídeo, o foco está na filmagem de um barco pequeno que navega da direita para a esquerda do vídeo. Logo após, ao cruzar com uma embarcação maior o foco do vídeo passa a ser o barco maior, navegando na direção oposta. O vídeo *Foreman* apresenta um homem filmando a si mesmo enquanto fala sobre uma obra e no final vira a câmera para a obra, mostrando uma construção. No *Walk* é apresentado um estudante descendo por uma escadaria enquanto conversa.

A Figura 17 apresenta a definição de cada sequência de vídeo utilizada para a análise.

Figura 17 – Sequências de vídeo.



Fonte: Do autor.

Esta sequência de vídeo é a filmagem de dois barcos que navegam em sentidos opostos em um rio. Optou-se por utilizar esta sequência porque ela apresenta um grande peso computacional para codificação de vídeo, uma vez que o vídeo inicia com foco em um barco menor navegando em uma direção e logo após passa para um barco maior navegando na direção oposta. Como os barcos estão em constante movimento, o cenário no fundo também possui movimentação quase constante. Na Figura 17 são apresentados alguns momentos da sequência de vídeo.

## 6.2 RESULTADOS

A análise de desempenho de cada software foi elaborada baseada no resultado da taxa de qualidade (PSNR), da complexidade de codificação, onde

adota-se como nível de complexidade o tempo necessário para a codificação do sinal de vídeo e da taxa de bits resultante de cada codificador.

Para um entendimento mais claro serão apresentados em gráficos os valores resultantes do processo de codificação da sequência de vídeo *Coastguard*, descrito no Capítulo anterior. Para a amostra de codificação foram utilizados 300 frames, sendo este o tamanho total da sequência do vídeo com diferentes taxas de quantização. Optou-se pela utilização dos resultados deste vídeo para demonstração gráfica porque nesta sequência tem-se o maior nível de complexidade computacional dentre as utilizadas como amostra, uma vez que a alteração entre as imagens de seus quadros é maior e constante.

A codificação dos vídeos utilizados para a análise de resultados foi feita em um computador com a seguinte configuração: Intel Core I5 3330, 3.0GHz e 8Gb de memória RAM com frequência de 1600MHz, onde é utilizado o Microsoft Windows 7 Profissional 64 bits.

Utilizou-se a variação de quantização para a coleta de resultados de codificação em diferentes níveis de qualidade, pois conforme apresentado por Medeiros (2011) e Souza (2011), por meio da variação de quantização é possível obter a maior taxa de compressão, embora também resulte na maior perda de qualidade resultante no processo de codificação. Deste modo o desempenho dos codificadores pode ser analisado com diferentes níveis de qualidade e compressão de cada sequência de vídeo.

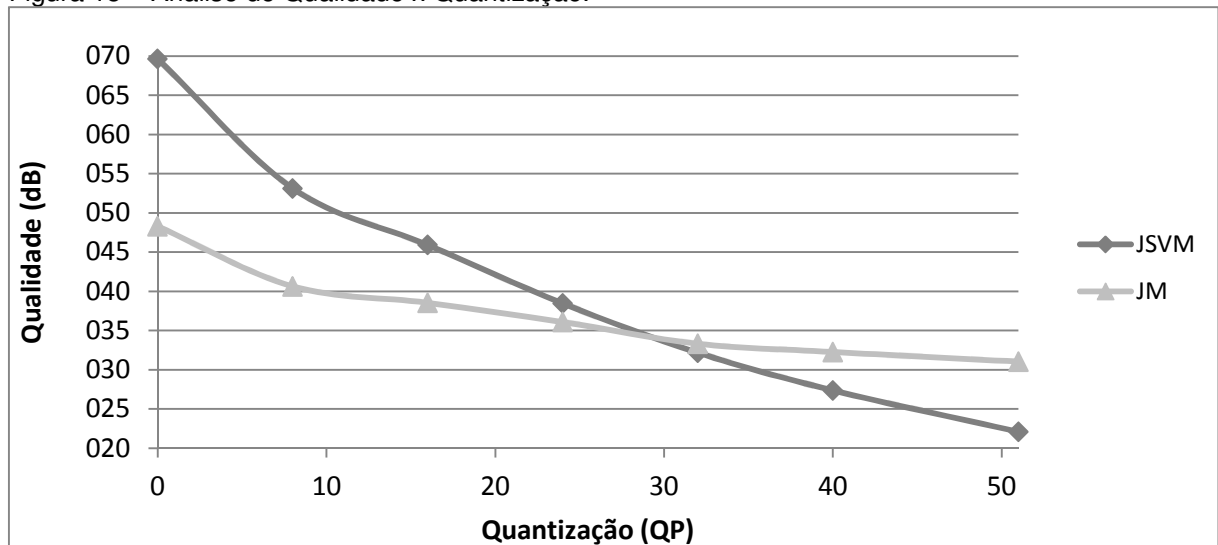
Para uma melhor representação do desempenho de cada codificador serão apresentados gráficos com os valores resultantes da codificação da sequência *Coastguard*, que representa o desempenho dos softwares na codificação do sinal de vídeo com maior custo computacional dentre os três estudados por esta pesquisa. A taxa de quantização deve variar entre 0 e 51, onde quanto maior for o valor, maior será a perda de qualidade aceitável para o vídeo. Para as simulações, foram utilizadas as seguintes taxas de quantização: 0, 8, 16, 32, 40 e 51. A definição de manter como valor fixo a taxa de quantização deu-se com base nos estudos de Souza (2011) em sua pesquisa acerca do desempenho do codificador H.264/SVC através da JSVM e Medeiros (2011) em seu relato sobre o desempenho do

codificador H.264/AVC através da JM, onde ambos utilizaram como base o mesmo ponto de partida.

### 6.2.1 Análise Comparativa de Qualidade x Quantização

Para demonstração da análise da qualidade serão adotados como padrão os valores de luminância Y, por ser o mais importante dentre os três, uma vez que os valores UV dependem diretamente deste. A Figura 18 apresenta o gráfico que expõe a relação de qualidade por quantização para cada um dos codificadores.

Figura 18 – Análise de Qualidade x Quantização.



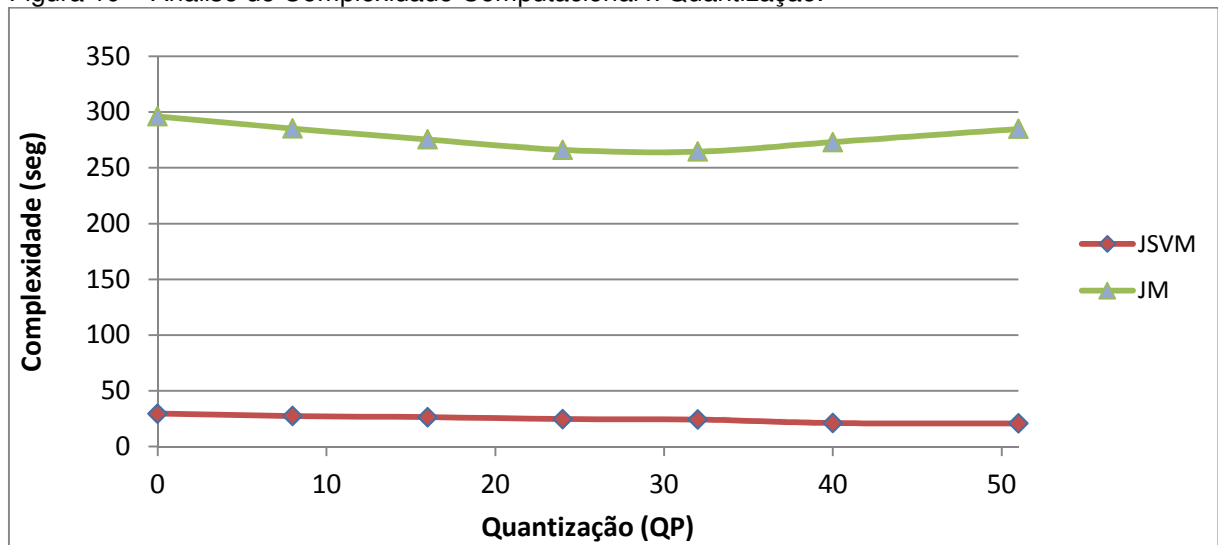
Fonte: Do autor.

Conforme apresentado na Figura 18, o codificador JSVM com taxas de quantização consegue atingir um alto nível de qualidade, embora com taxas de quantização maiores a perda de qualidade dos algoritmos da JSVM seja maior.

### 6.2.2 Taxa de Quantização x Complexidade

Outro ponto importante a ser analisado em um estudo comparativo entre os dois codificadores é o nível de complexidade computacional de cada um. Seguindo ainda como base a taxa de quantização, é possível observar na Figura 19 o gráfico que relaciona estes itens.

Figura 19 – Análise de Complexidade Computacional x Quantização.



Fonte: Do autor.

Na Figura 19 fica claro que a complexidade do codificador JM é maior que a do JSVM, sendo este um dos pontos mais relevantes entre o desempenho de ambos. Esta diferença se dá pela remodelagem dos algoritmos de ME e MC implementados para o JSVM.

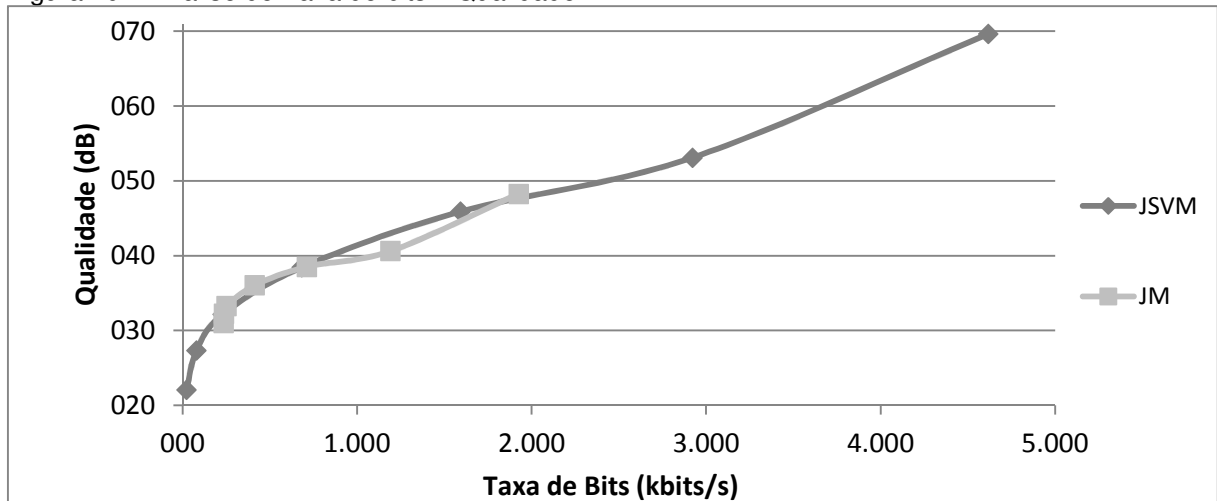
### 6.2.3 Análise de Taxa de Bits x Qualidade

O gráfico apresentado na Figura 20 demonstra a relação entre qualidade e taxa de bits resultantes no processo de codificação com cada um dos softwares de referência. Como a taxa de *bits* é diretamente proporcional à qualidade da imagem, pois uma imagem de melhor qualidade exige mais informações a serem processadas, é possível observar que o sinal gerado pelo codificador JSVM por atingir níveis de qualidade mais elevados também gera uma taxa de bits mais elevada, porém como um índice de qualidade variando entre 30~40 dB é considerado um bom resultado, pode-se dizer que em questão de qualidade x taxa de *bits* ambos os codificadores possuem resultados compatíveis.

Uma característica bastante marcante apresentada nos gráficos, é a divergência entre o nível de qualidade atingido e a taxa de bits gerada por cada codificador. Isto se dá pelo fato de que no formato dos vídeos utilizados nos testes (QCIF), a partir do momento em que a taxa de bits atinge valores muito elevados,

passa a não utilizar mais todo o recurso do codificador, pois para a transmissão passa a não ser mais necessária codificação da sequência de vídeo.

Figura 20 – Análise de Taxa de bits x Qualidade.

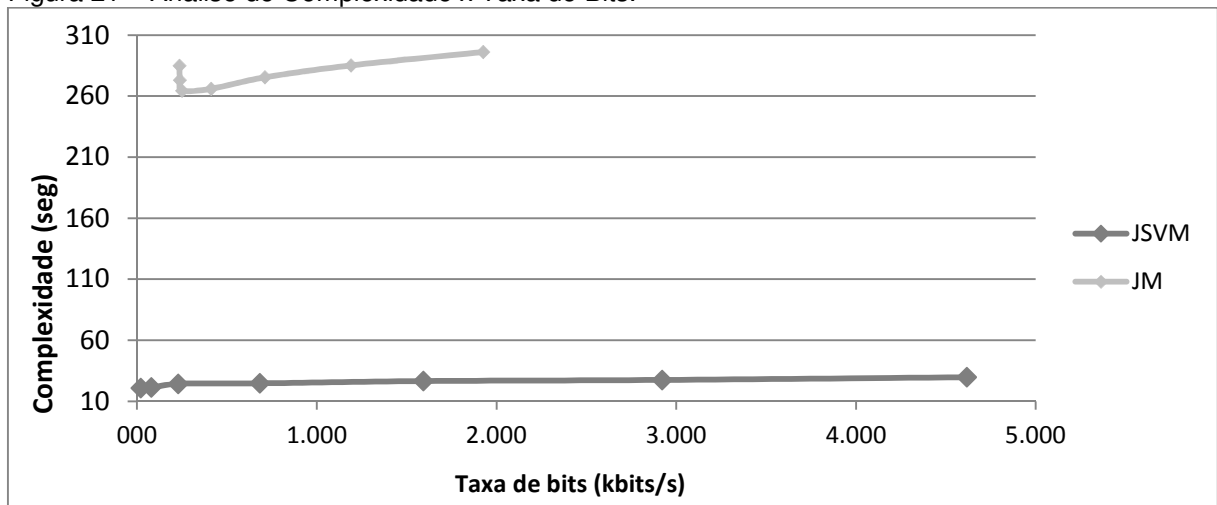


Fonte: Do autor.

#### 6.2.4 Análise da Complexidade x Taxa de Bits

Neste item será demonstrada a análise da complexidade de cada codificador relacionada com a taxa de *bits* gerada por cada um. Na Figura 21 é possível notar uma significativa diferença entre a complexidade computacional entre os codificadores

Figura 21 – Análise de Complexidade x Taxa de Bits.



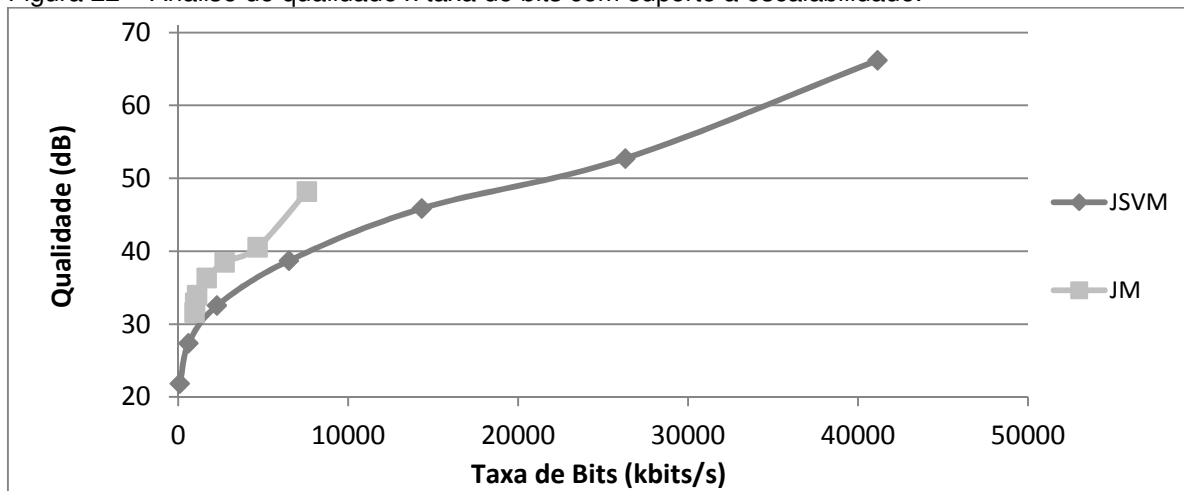
Fonte: Do autor.

### 6.2.5 Análise do Resultado de Codificação com Escalabilidade

Como o SW JM não possui suporte à escalabilidade, para que seja possível realizar uma análise de resultados a ser comparada com o sinal gerado pela JSVM com suporte a escalabilidade, foram retiradas amostras da JM codificando um sinal de vídeo no formato CIF, onde no codificador JSVM também foi gerado um resultado escalável para este formato. Para que fique mais fácil compreender o ganho de desempenho obtido com o SW JSVM, é possível observar os valores apresentados nas figuras 22 e 23, que apresentam as relações entre qualidade resultante, taxa de bits e complexidade computacional para cada codificador.

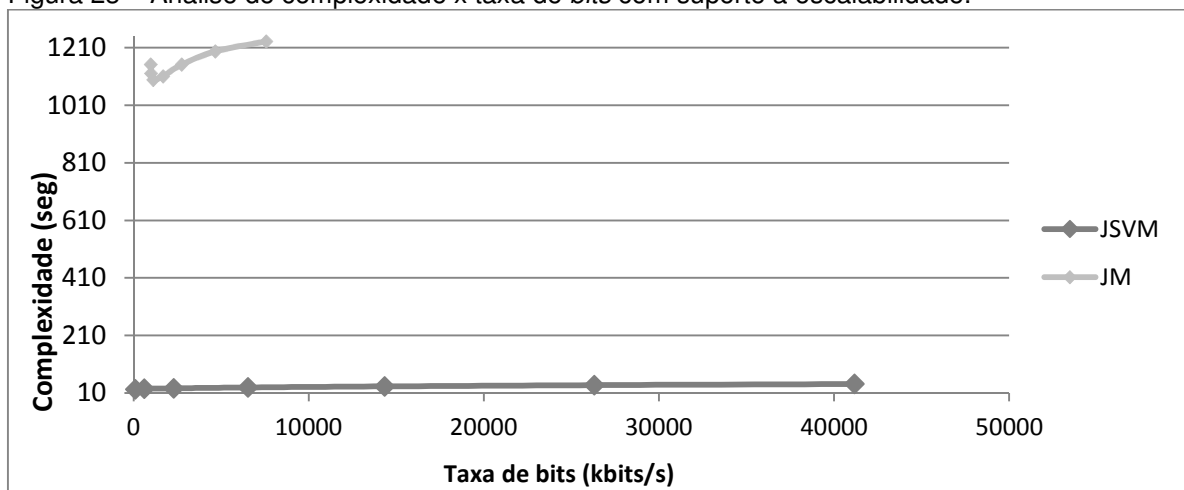
Deve ser lembrado que o resultado da codificação pela JSVM trabalha com uma taxa de bits mais elevada porque o vídeo escalável possui suporte a resolução CIF e QCIF no mesmo *bit rate*, uma vez que parte da informação utilizada para a execução do vídeo no formato CIF provém do sinal gerado pelo codificador ao codificar o vídeo no formato QCIF.

Figura 22 – Análise de qualidade x taxa de *bits* com suporte a escalabilidade.



Fonte: Do autor.

Figura 23 – Análise de complexidade x taxa de *bits* com suporte a escalabilidade.



Fonte: Do autor.

Com base nos valores apresentados nos gráficos, é possível percebermos que o codificador JSVM em todos os testes apresentou uma complexidade computacional significativamente reduzida, embora para alguns casos a taxa de bits resultante da codificação tenha sido um pouco superior a do JM. Outro quesito bastante importante, é que devido ao aprimoramento dos algoritmos de estimação e compensação de movimento, o codificador JSVM consegue atingir níveis de qualidade mais elevados.

## 7 Conclusão

Este trabalho apresenta uma descrição sobre o que é e como funciona um vídeo digital. Conceitos estes, que são necessários para que seja possível obter um entendimento sobre o funcionamento dos softwares de referência abordados pelo foco principal desta obra, que é a análise comparativa do funcionamento e eficiência dos padrões de codificação de vídeo H.264/AVC e H.264/SVC que abrange o suporte a escalabilidade espacial e temporal.

Para a extração dos resultados de codificação, foi utilizado o software de referência JM para vídeos sem suporte a escalabilidade e o software de referência JSVM para vídeos com suporte à escalabilidade. Para ambos os casos foram efetuados testes com três sequências de vídeo comumente utilizadas na avaliação de desempenho de codificação de vídeo, são elas: *Foreman*, *Coastguard* e *Walk*. Com a codificação destes três vídeos tem-se um bom cenário de resultados uma vez que neles estão presentes situações de diferentes complexidades computacionais, onde cada um possui um nível diferente de alteração entre seus quadros durante a execução do vídeo.

Ao ser feita a análise dos resultados obtidos nos testes, e demonstrados através dos gráficos no Capítulo 6, pode ser identificado que os sinais resultantes da codificação com o software de referência JM podem apresentar um ganho de qualidade com relação ao resultante da codificação pela JSVM, porém o custo computacional dos algoritmos da JM são consideravelmente maiores, resultando em uma grande diferença de desempenho entre os codificadores.

Com base nas informações extraídas dos testes, ficou claro que a JSVM possui um desempenho superior à JM com relação ao tempo de codificação e em algumas situações também em qualidade, além de apresentar suporte a escalabilidade em diversas características da sequência de vídeo, sendo assim preferível em situações que exijam transmissões em tempo real ou envio de sinais para dispositivos com suporte a diferentes resoluções.

Para pesquisas futuras, tem-se a opção de estudar o desempenho da codificação de vídeos com e sem suporte a escalabilidade utilizando somente a

JSVM, ou ainda uma análise de desempenho do codificador H.264/SVC para vídeos 3D.

## REFERÊNCIAS

AGOSTINI, Luciano Volcan. **Desenvolvimento de Arquiteturas de Alto Desempenho Dedicadas a Compressão de Vídeo Segundo o Padrão H.264/AVC**. 2007. 172 f. Tese (Doutorado) - Universidade Federal do Rio Grande do Sul, Porto Alegre, 2007.

ALDANA, Santiago; GRISALES, Juan Camilo; REINA, Jackson. Análisis del estado del arte de la codificación de vídeo. **Ingenium**: Revista de la Facultad de Ingeniería, Bogotá, n. , p.58-73, jun. 2011.

AMORIM, Francisco Borges de Pina. **CHAVEAMENTO DE STREAMS DE VÍDEO CODIFICADAS NO PADRÃO H.264/AVC**. 2009. 40 f. Tese (Graduação) - Universidade de Brasília, Brasília, 2009.

DARONCO, Leonardo Crauss, **Avaliação Subjetiva de Qualidade Aplicada à Codificação de Vídeo Escalável**, Dissertação para obtenção do grau de Mestre em Ciência da Computação, Porto Alegre, 2009.

DUMA, Luiz H.; FONSECA, Keiko V. O.; POHL, Alexandre de A. P.. **Técnicas de otimização para codificadores H.264**. Curitiba, 2010.

FONSECA, C. M. **Investigação em software sobre o padrão H.264 escalável de compressão de vídeos**. Pelotas, 2008.

GONZALEZ, Rafael C.; WOODS, Richard E.. **Processamento de Imagens Digitais**. São Paulo - Sp: Edgard Blücher Ltda., 2003.

GRECO, Felipe Silveira. **REDES NEURAIS PARA RECONHECIMENTO DE PADRÕES DE VÍDEO MPEG4**. 2011. 37 f. Trabalho de Conclusão de Curso de Engenharia Elétrica (3º Grau) - Universidade Federal do Paraná, Curitiba, 2011.

INGLIS, A. F. ; Luther, A. **Video engineering**. McGraw-Hill, 1996.

ITU-T. Recommendation H.264. **Advanced Video Coding for Generic Audiovisual Services**. Int'l Telecomm. Union, Geneva, Switzerland, 2005.

ITU-T & ISO/IEC JVT. **H.264/AVC Reference Software Encoder version JM 18.0**. Disponível em: <<http://iphome.hhi.de/suehring/tml/>>. Acesso em: 24 Maio 2013.

ITU-T & ISO/IEC JVT. **H.264/SVC Reference Software Encoder version JSVM 9.19.14**. Disponível em: <<http://iphome.hhi.de/suehring/tml/>>. Acesso em: 24 Maio 2013.

KOWN, Soon-Kak; TAMHANKAR, A.; RAO, K. R.. **Overview of H.264;MPEG-4 part 10**. Journal of Visual Communication & Image Representation, Revisão 17, p. 186-216, Agosto 2005.

MEDEIROS, Tiago Pizzetti. **Segmentação Quadtree para Seleção de Blocos na Codificação de Vídeo H.264/AVC**. 2011. 121 f. Trabalho de Conclusão de Curso (Bacharel) - Universidade do Extremo Sul Catarinense - Unesc, Criciúma, 2011.

PORTO, Marcelo Schiavon. **Desenvolvimento Algorítmico e Arquitetural para Estimação de Movimento na Compressão de Vídeo de Alta Definição**. 2012. 166 f. Tese (Doutorado) - Curso de Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2012.

RAVI, Aruna. **Performance Analysis and Comparison of Dirac Video Codec With H.264/MPEG-4 Part 10 AVC**. 2009. 117 f. Tese (Mestrado) - Curso de Ciência e Engenharia Elétrica, The University Of Texas At Arlington, Arlington, 2009.

RICHARDSON, I. **H.264/AVC and MPEG-4 Video Compression – Video Codinf for Next-Generation Multimedia**. Chichester: John Wiley and Sons, 2003.

ROSA, Leandro Zanetti Paiva da. **Investigação sobre Algoritmos para a Estimação de Movimento na Compressão de Vídeos Digitais de Alta Definição: Uma Análise Quantitativa**. 2007. 155 f. Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Pelotas, Pelotas, 2007.

SCHWARZ, H.; MARPE, D.; WIEGAND, T. **Overview of the Scalable Video Coding Extension of the H.264/AVC Standard**. IEEE Transactions on Circuits and Systems for Video Technology, Vol. 17, No. 9, p. 1103-1120, Sept. 2007.

SEGALL, C.A.; SULLIVAN, G.J. **Spatial Scalability Within the H.264/AVC Scalable Video Coding Extension**. IEEE Transactions on Circuits and Systems for Video Technology, Vol. 17, No. 9, pp. 1121-1135. Sept. 2007.

SILVA, André Marcelo Coelho da. **Um Estudo Sobre o Padrão H.264/AVC de Compressão de Vídeo**. 2007. 44 f. Dissertação (Pós-graduação) - Universidade Católica de Pelotas, Pelotas, 2007

SOARES, Luiz Fernando Gomes. **Fundamentos de Sistemas Multimídia: Part 1 - Aquisição, Codificação e Exibição de Dados**. 2007. 40 f. Monografia (Graduação) -

Curso de Ciência da Computação, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2007.

SOLER, Luciano. **Compactação de Vídeo Escalável**. 2006. 186 f. Dissertação (Mestrado) - Universidade Federal do Rio Grande do Sul, Porto Alegre, 2006.

SOUZA, Diego Felix de. **CODIFICAÇÃO DE VÍDEO VIA CASAMENTO DE PADRÕES USANDO ESTIMAÇÃO DE MOVIMENTO**. 2010. 88 f. Dissertação (Mestrado) - Curso de Programa de Pós-graduação em Engenharia Elétrica, Coppe, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2010.

SOUZA, Márlon Henrique de. **Avaliação de Desempenho da Escalabilidade Espacial na Recomendação H.264/SVC para Codificação de Vídeo**. 2011. 93 f. Trabalho de Conclusão de Curso (Bacharel) - Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense - Unesc, Criciúma, 2011.

WIEGAND, T.; SULLIVAN, G.; REICHEL, J.; SCHWARZ, H.; WIEN, M. **Joint draft 11 of SVC amendment, joint video team, Doc. JVT-X201**, Jul. 2007.

WIEN, Mathias; SCHWARZ, Heiko; OELBAUM, Tobias, **Performance Analysis of SVC**. IEEE Transactions on Circuits and Systems for Video Technology, Vol. 17, No. 9, p. 1194-1203, Sept. 2007

Wikipedia. **Televisão digital no Brasil**. Disponível em: <[http://pt.wikipedia.org/wiki/Televis%C3%A3o\\_digital\\_no\\_Brasil](http://pt.wikipedia.org/wiki/Televis%C3%A3o_digital_no_Brasil)>. Acesso em: 25 Maio 2010.

Wikipedia. **MPEG-4**. Disponível em: <<http://pt.wikipedia.org/wiki/MPEG-4>>. Acesso em: 25 Maio 2010.

**APÊNDICE**

# Análise Comparativa de Desempenho Entre os Codificadores de Vídeo H.264/AVC JM E H.264/SVC JSVM

Lauro A. Nascimento<sup>1</sup>, Evânio Ramos Nicoleit<sup>2</sup>

<sup>1</sup>Universidade do Extremo Sul Catarinense (UNESC) – Criciúma – SC – Brasil

<sup>2</sup> Universidade do Extremo Sul Catarinense (UNESC) – Criciúma – SC – Brasil

{laaauro,evaniorn}@gmail.com

**Abstract.** *This article aims at presenting the concept approached by H.264/AVC and H.264/SVC encoders, well as presenting a study on the performance shown for each encoder.*

**Resumo.** *Este artigo tem por objetivo apresentar o conceito abordado pelos codificadores H.264/AVC e H.264/SVC, bem como apresentar um estudo sobre o desempenho apresentado por cada codificador.*

## 1. Introdução

A transmissão de vídeo está cada dia mais presente no cotidiano de todos. Com as novas tecnologias de imagens digitais, cada vez mais se faz necessária uma solução capaz de permitir a transmissão de imagens em alta definição através de sinais limitados. Porém, para que seja possível esta transferência sem que sejam apresentadas imagens travadas ou distorcidas enfrenta-se um grave problema: imagens de alta definição exigem uma grande quantidade de informação para serem transferidas.

É neste contexto que esta pesquisa se enquadra, realizando uma análise comparativa entre os dois padrões de compressão de vídeo que representam o estado da arte na área de codificação de vídeo da atualidade, afim de obter-se uma estimativa através da análise de resultados obtidos pelo software de referência de cada codificador qual se torna mais vantajoso de acordo com as necessidades dos usuários.

## 2. Vídeo Digital

Amorim (2009) apresenta o conceito de vídeo digital como sendo uma sequência de imagens exibidas a tal velocidade que passem para o olho humano a ilusão de movimento contínuo. Amorim também descreve que imagens são formadas por pontos, denominados pixels. Partindo deste ponto de vista, podemos então deduzir que um vídeo digital é uma sequência de pixels que quando exibidos apresentam imagens capazes de causar ao olho humano a sensação de movimento.

De um ponto de vista semelhante ao supra deduzido, Souza (2010) explana em sua tese de mestrado que um vídeo digital pode ser representado como um sinal tridimensional, sendo uma dimensão de tempo e duas de espaço, onde este sinal seria constituído por pontos de luminância e crominância.

### 3. Compressão de Video Digital

Como descrito por Gonzalez, Woods (2003, p. 219), “o termo *compressão de dados* refere-se ao processo de redução da quantidade de dados necessária para representar certa quantidade de informação”, e, sem o emprego de técnicas de compressão de vídeo, hoje seria completamente inviável a transmissão de imagens em tempo real por qualquer que fosse o meio de comunicação.

De acordo com Greco (2011), o princípio da compressão de vídeo é a eliminação de redundâncias do sinal do mesmo. Amorim (2009), em seu estudo à respeito de codificação de sinais, salienta três tipos de redundâncias que podem ser trabalhadas para uma efetiva compressão de vídeo e que são utilizadas pelos codificadores H.264/AVC e H.264/SVC: Redundância Temporal e Espacial, Redundância em Entropia e Redundância psico-visual.

#### 3.1. Redundância Temporal e Espacial

No processo de codificação de vídeo, os pixels são separados em grupos denominados macroblocos, onde são extraídos os valores de luminância e crominância para cada macrobloco. Codificando os sinais de luminância e crominância de cada macrobloco, a identificação de outros macroblocos com sinais semelhantes caracteriza a redundância. Trabalhando-se somente com o *bitstream* gerado pela codificação de um macrobloco, agregado ao posicionamento de cada macrobloco que apresente um sinal semelhante é gerada uma significativa compressão do sinal de vídeo gerado pelo codificador (Rosa, 2007).

Para eliminação de redundância temporal, é avaliada a capacidade de distinção entre as imagens pela visão humana. A captação de imagens feita por uma câmera de vídeo pode chegar a centenas de imagens por segundo, porém para que se tenha um resultado satisfatório, a sucessão deve ficar entre 24 a 30 quadros por segundo para que o sistema visual humano perceba a sequência de quadros como um vídeo contínuo (Amorim, 2009).

Um processo amplamente utilizado para compressão de vídeo baseando-se na redundância temporal e espacial é a Estimção de Movimentos (EM). A EM é responsável por encontrar uma correlação entre quadros, mapeando a redundância temporal entre os quadros vizinhos de uma cena visando codificar apenas os macroblocos que se alteram entre um quadro e outro (Porto, 2012).

#### 3.2. Redundância em Entropia

A codificação em entropia é a responsável pela compressão a partir da redundância gerada no sinal dos quadros codificados. O objetivo principal é representar os dados de maior ocorrência na sequência pela menor representação binária possível (Duma; Fonseca; Pohl, 2010). De acordo com o explicado por Porto (2012), a quantidade de informações geradas em um símbolo para transmissão é diretamente proporcional a probabilidade de ocorrência daquele símbolo. Com isto, símbolos que venham a aparecer muitas vezes deverão ser transmitidos com menos bits.

#### 3.3. Redundância Psico-Visual

A redundância psico-visual aborda um quesito diferente das demais descritas. “[...] Ao contrário da codificação e da redundância interpixels, a redundância psico-visual é associada com a informação visual quantificável ou real” (GONZALEZ; WOODS, 2003, p. 224). Considerando as limitações apresentadas pela visão humana, após a codificação dos sinais de

luminância e crominância dos macroblocos apresentados nas imagens, pequenas variações dos mesmos podem ser descartadas sem causar perdas de conteúdo relevantes para o olho humano, porém que podem trazer uma significativa redução ao tamanho final do arquivo.

#### 4. Codificador de Vídeo H.264/AVC

O codificador é o algoritmo que trabalha com os quadros do vídeo retirando as redundâncias de acordo com o citado anteriormente. A descrição do funcionamento do codificador será apresentada tendo como base o descrito por Agostini (2007), onde o mesmo explana o funcionamento do codificador do padrão H.264/AVC.

“Uma característica interessante do padrão H.264/AVC é que apenas o decodificador é normatizado. Então vários graus de liberdade podem ser explorados na implementação do codificador” (Agostini, 2007, p. 47). Ainda segundo Agostini (2007), como o codificador não é normatizado, torna-se livre a escolha dos algoritmos a serem utilizados no mesmo, desde que no final seja gerado um *bitstream* compatível com padrão entendido pelo decodificador.

Os quadros a serem processados são divididos em dois tipos, I e P. Os quadros do tipo I são utilizados para sincronizar o sinal do vídeo, e só podem ser comprimidos aproveitando-se a redundância espacial (compressão intra-quadro), sem que as informações presentes nos quadros vizinhos influenciem no processo de compressão do mesmo. Isto se dá porque outros quadros serão construídos a partir deste, e, devido a isto, o máximo de informações devem ser preservadas. Os quadros do tipo P são os quadros resultantes do processo de codificação a partir dos quadros antecedentes. Para o processamento de quadros do tipo P para P, são utilizados além dos algoritmos de compressão intra-quadros, mais técnicas de eliminação de redundância temporal, realizando assim a codificação inter-quadros. Estas técnicas são divididas normalmente em duas etapas chamadas estimação de movimento e compensação de movimento (Agostini, 2007).

De modo mais detalhado, o processo de codificação se inicia pelo módulo de Predição Inter-quadros, onde estão presentes os blocos de estimação de movimento (ME – *Motion Estimation*) e compensação de movimento (MC – *Motion Compensation*). Nestes dois blocos estão presentes as maiores inovações e os maiores ganhos obtidos pelo padrão H.264/AVC em relação aos demais padrões comumente utilizados. Porém, em virtude disso, estão também contidos os algoritmos de maior complexidade de toda a estrutura do codificador e decodificador. Mais adiante será explicado com maiores informações o funcionamento de cada um destes blocos.

Depois de feito os tratamentos de EM e MC, é feita a subtração dos valores do bloco original com os valores resultantes do processo de codificação. A informação resultante desta operação é chamada de resíduo. “O resíduo é a diferença dos elementos de luminância entre o quadro de referência e o quadro de vídeo atual (que está sendo codificado) [...]” (Silva, 2007, p. 13).

A transformada (bloco T do diagrama de quadros apresentado na Figura 1) recebe o resíduo e é responsável pela eliminação das informações que não são relevantes para o olho humano, ou seja, retirar as informações que o olho humano não é capaz de distinguir das imagens. Neste bloco são reduzidas as redundâncias espaciais que permaneçam após o processo de codificação inter-quadros. Além disso, a transformada também é responsável pela conversão dos valores dos resíduos que estão no domínio espacial para o domínio de frequência, que podem obter resultados mais satisfatórios com os algoritmos de codificação de entropia, presentes do módulo de quantização.

No padrão H.264/AVC foram introduzidas algumas inovações para o módulo da transformada. Desde os perfis *Baseline*, *Main* e *Extended*, as dimensões da transformada foram definidas em 4x4. Nos perfis High, foi adicionado a capacidade de adaptação entre dimensões 4x4 ou 8x8. Outra inovação, é que uma segunda transformada é aplicada sobre os elementos resultantes da DCT para todos os blocos de crominância e para os macroblocos em que é feita a predição intraquadros 16x16. Ainda para a transformada, foi adicionada uma implementação em hardware de ponto fixo para evitar erros de casamento entre o codificador e o decodificador.

Para codificação de entropia o H.264/AVC foram adicionadas duas ferramentas que aumentam bastante a eficiência de codificação, que são os métodos CAVLC e CABAC. Para a codificação de entropia, a principal inovação introduzida na codificação aritmética é o uso de codificação adaptativa baseada em contextos. Neste método, o modo como são codificados os elementos sintáticos depende do elemento a ser codificado, da fase em que se encontra o algoritmo de codificação e dos elementos sintáticos que já foram codificados (Silva, 2007).

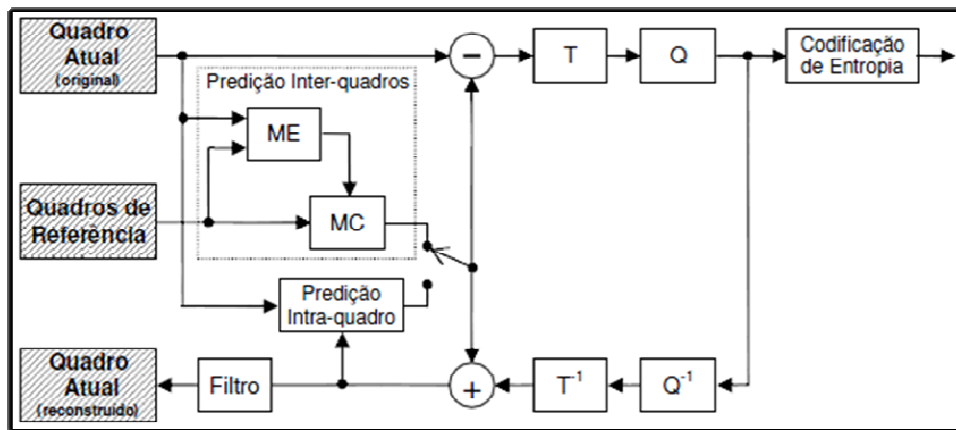


Figura 1. Núcleo do codificador H.264/AVC

## 5. Codificador H.264/SVC

O padrão H.264/AVC possui uma extensão desenvolvida que visa o suporte a escalabilidade denominado H.264/SVC – *Scalable Video Coding* (Wiegand et al., 2007). Além do suporte a escalabilidade, tem-se como melhorias desenvolvidas pelo H.264/SVC o uso de técnicas de predição de movimentos mais avançadas, aumentando assim a eficiência da codificação; redução na complexidade de codificação e a utilização do próprio H.264 para codificação, que resulta em um considerável ganho de desempenho na codificação de vídeo. De acordo com os estudos apresentados por Daronco (2009), a primeira apresentação do H.264/SVC foi feita em setembro de 2007. Este novo padrão apresentava a solução para os problemas enfrentados até então com a codificação de vídeo escalável.

Soler (2006) descreve a escalabilidade como sendo a capacidade de através do decodificador se extrair de um *bitstream* subconjuntos de *bits* com variações de alguma característica da imagem ou vídeo, como por exemplo, resolução espacial ou qualidade de imagem. Daronco (2009) descreve que, apesar de o termo escalabilidade possuir uma abrangência bem grande quanto a seu significado, para o padrão H.264 é adotada a ideia de adaptação do fluxo de vídeo as necessidades e capacidades do usuário, de modo com que seja possível a remoção de partes do fluxo do vídeo para que as necessidades sejam mais bem atendidas.

Isto significa que O H.264/SVC provê suporte a múltiplas resoluções de exibição com um único *bitstream*. Adicionalmente, o H.264/SVC também suporta escalabilidade temporal, que está relacionada com a taxa de quadros, ou seja, alguns quadros podem ser descartados; e a escalabilidade qualitativa, que é o enriquecimento na fidelidade dos quadros mantendo-se a resolução entre as camadas. O H.264/SVC também permite a combinação dessas técnicas (SEGALL; SULLIVAN, 2007). Na Figura 2 é possível observar o diagrama do núcleo do padrão H.264/SVC.

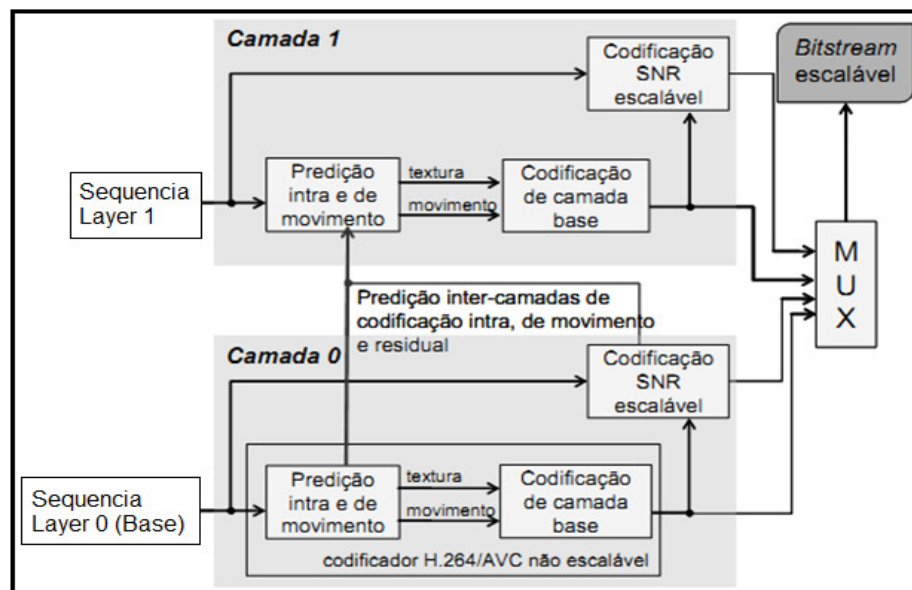


Figura 2. Núcleo do codificador H.264/SVC

Wien, Schwarz e Oelbaum (2007), em sua análise de performance do H.264/SVC, puderam concluir que este padrão não apresentou perdas de desempenho relevantes com relação ao padrão H.264/AVC, agregando as características presentes em métodos de codificação de vídeo escalável com significativa redução de complexidade.

## 6. Comparação dos Codificadores

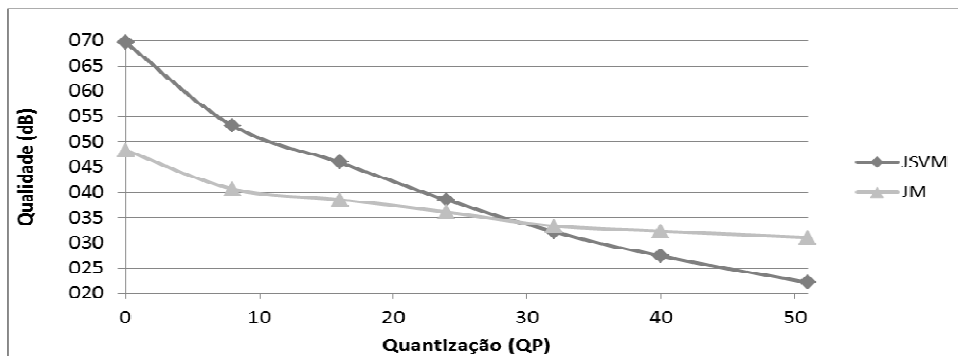
A extração de dados para a análise a ser efetuada nesta pesquisa foi feita através dos *softwares* de referência para codificação de vídeo H.264 com e sem suporte a escalabilidade descritos anteriormente. Os dois compiladores foram construídos utilizando-se o Microsoft Visual Studio 2012, em um computador com a seguinte configuração: Intel Core I5 3330, 3.0GHz e 8Gb de memória RAM com frequência de 1600MHz, onde é utilizado o Microsoft Windows 7 Professional 64 bits.

Os dados utilizados na comparação de resultados dos codificadores serão extraídos das sequências de vídeo *Coastguard*, *Foreman* e *Walk*, comumente utilizadas na análise de desempenho de codificação de vídeo. Para a análise dos resultados sem escalabilidade, foram utilizados vídeos no formato QCIF, e para a comparação de resultados utilizando o suporte a escalabilidade do SW JSVM foram utilizados vídeos no formato CIF. Optou-se por utilizar estas três sequências de vídeo por apresentarem um bom nível de abrangência nos testes, uma vez que cada uma apresenta um diferente nível de complexidade e de informações a serem processadas pelo codificador.

Utilizou-se a variação de quantização para a coleta de resultados de codificação em diferentes níveis de qualidade, pois conforme apresentado por Medeiros (2011) e Souza (2011), por meio da variação de quantização é possível obter a maior taxa de compressão, embora também resulte na maior perda de qualidade resultante no processo de codificação. Deste modo o desempenho dos codificadores pode ser analisado com diferentes níveis de qualidade e compressão de cada sequência de vídeo.

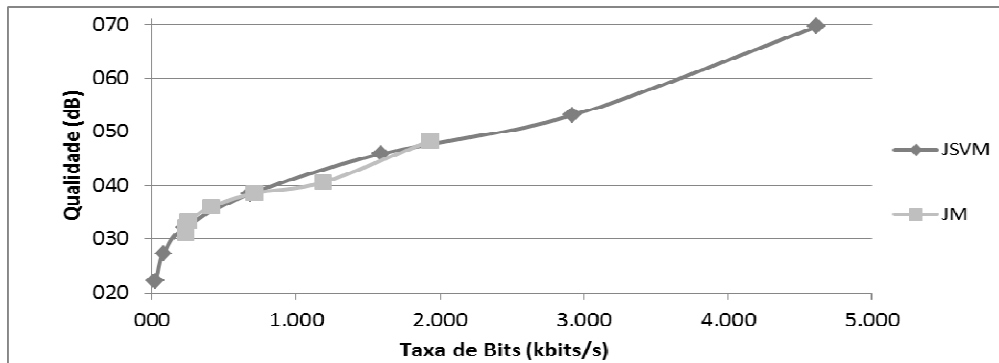
Para uma melhor representação do desempenho de cada codificador serão apresentados gráficos com os valores resultantes da codificação da sequência *Coastguard*, que representa o desempenho dos softwares na codificação do sinal de vídeo com maior custo computacional dentre os três estudados por esta pesquisa. A taxa de quantização deve variar entre 0 e 51, onde quanto maior for o valor, maior será a perda de qualidade aceitável para o vídeo. Para as simulações, foram utilizadas as seguintes taxas de quantização: 0, 8, 16, 32, 40 e 51. A definição de manter como valor fixo a taxa de quantização deu-se com base nos estudos de Souza (2011) em sua pesquisa acerca do desempenho do codificador H.264/SVC através da JSVM e Medeiros (2011) em seu relato sobre o desempenho do codificador H.264/AVC através da JM, onde ambos utilizaram como base o mesmo ponto de partida.

Para demonstração da análise da qualidade serão adotados como padrão os valores de luminância Y, por ser o mais importante dentre os três, uma vez que os valores UV dependem diretamente deste. Na Figura 3, onde é apresentado o gráfico que expõe a relação de qualidade por quantização para cada um dos codificadores, nota-se que o nível de qualidade obtido através do codificador JM apresenta uma variação menor e com uma taxa de quantização maior possui menor perda de qualidade, porém utilizando-se o JSVM com taxa de quantização menor é possível obter-se um nível de qualidade superior.



**Figura 3. Análise de Qualidade x Quantização.**

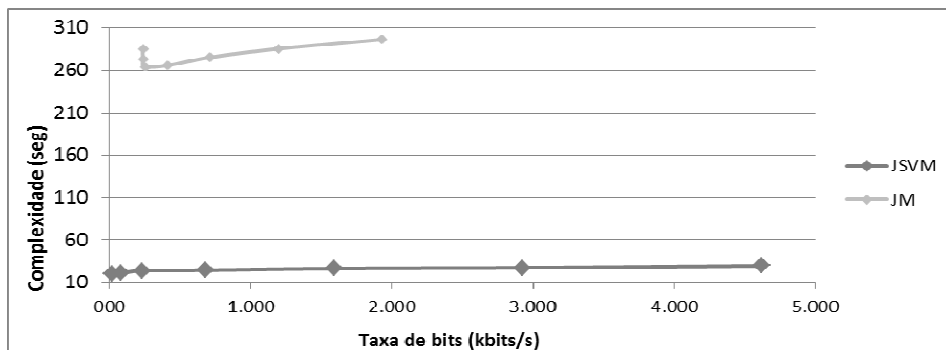
Sabendo-se que o nível de qualidade obtida pelo processo de codificação a partir de cada software pode variar, para que a análise de desempenho possa ser mais precisa, é importante que seja também observada a taxa de bits resultante no processo de codificação de cada software. A comparação de resultados de taxa de bits com relação à qualidade do sinal gerado está apresentada na Figura 4.



**Figura 4. Análise de Taxa de bits x Qualidade.**

De acordo com os valores apresentados no gráfico, é possível visualizar que a taxa de bits resultante do processo de codificação com cada software é semelhante, porém como o JSVM atinge níveis mais altos de fidelidade ao vídeo original, para estes casos resulta também em uma taxa de bits igualmente maior.

Outro ponto analisado por esta pesquisa na comparação dos codificadores, foi a complexidade computacional envolvida no processo de codificação, onde entende-se por complexidade o tempo necessário para o término da codificação de cada sequência de vídeo. Na figura 5 temos os resultados da análise de complexidade de codificação da sequência Coastguard.



**Figura 5. Análise de Complexidade x Taxa de Bits.**

Visualizando os resultados apresentados no gráfico da Figura 5, nota-se que o codificador JSVM apresenta uma complexidade computacional significativamente inferior. Esta diferença de resultados se dá pela reestruturação dos algoritmos efetuada durante o desenvolvimento do JSVM.

Como o SW JM não possui suporte à escalabilidade, para que seja possível realizar uma análise de resultados a ser comparada com o sinal gerado pela JSVM com suporte a escalabilidade, foram retiradas amostras da JM codificando um sinal de vídeo no formato CIF, onde no codificador JSVM também foi gerado um resultado escalável para este formato. Para que fique mais fácil compreender o ganho de desempenho obtido com o SW JSVM, é possível observar os valores apresentados nas figuras 6 e 7, que apresentam as relações entre qualidade resultante, taxa de bits e complexidade computacional para cada codificador.

Deve ser lembrado que o resultado da codificação pela JSVM trabalha com uma taxa de bits mais elevada porque o vídeo escalável possui suporte a resolução CIF e QCIF no mesmo *bit rate*, uma vez que parte da informação utilizada para a execução do vídeo no formato CIF provém do sinal gerado pelo codificador ao codificar o vídeo no formato QCIF.

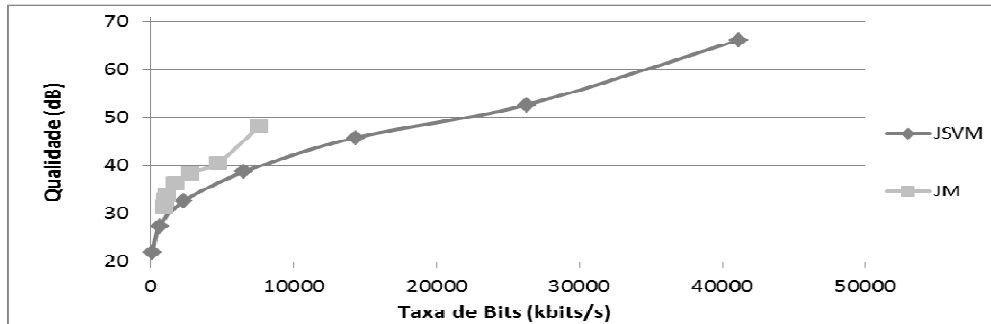


Figura 6. Análise de qualidade x taxa de *bits* com suporte a escalabilidade.

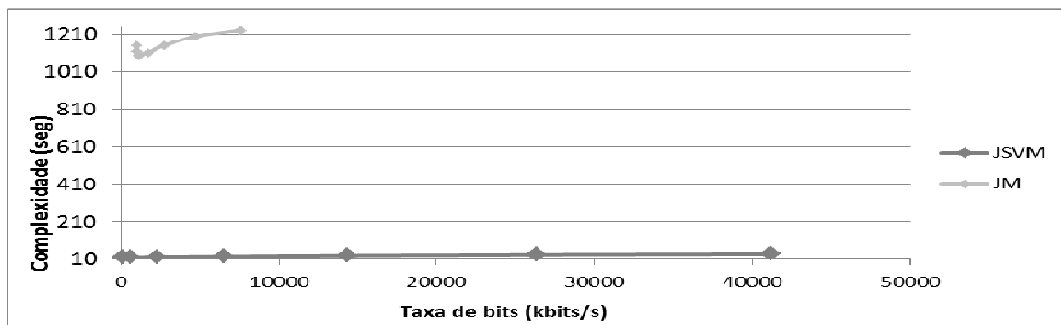


Figura 7. Análise de complexidade x taxa de *bits* com suporte a escalabilidade.

Com base nos valores apresentados nos gráficos, é possível percebermos que o codificador JSVM em todos os testes apresentou uma complexidade computacional significativamente reduzida, embora para alguns casos a taxa de bits resultante da codificação tenha sido um pouco superior a do JM. Outro quesito bastante importante, é que devido ao aprimoramento dos algoritmos de estimação e compensação de movimento, o codificador JSVM consegue atingir níveis de qualidade mais elevados.

#### 4. Conclusão

Ao ser feita a análise dos resultados obtidos nos testes, e demonstrados graficamente no corpo deste artigo, pode ser identificado que para casos específicos os sinais resultantes da codificação com o software de referência JM podem apresentar um ganho de qualidade com relação ao resultante da codificação pelo JSVM, porém o custo computacional dos algoritmos do JM são consideravelmente maiores, resultando em uma significativa diferença de desempenho entre os codificadores.

Com base nas informações extraídas dos testes, ficou claro que o JSVM possui um desempenho superior ao JM com relação ao tempo de codificação e em grande parte das simulações em qualidade; além de apresentar suporte a escalabilidade em diversas características da sequência de vídeo, sendo assim preferível em situações que exijam transmissões em tempo real ou envio de sinais para dispositivos com suporte a diferentes resoluções.

## Referências

- Agostini, Luciano Volcan. Desenvolvimento de Arquiteturas de Alto Desempenho Dedicadas a Compressão de Vídeo Segundo o Padrão H.264/AVC. 2007. 172 f. Tese (Doutorado) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2007.
- Amorim, Francisco Borges de Pina. Chaveamento de Streams de Vídeo Codificadas no Padrão H.264/AVC. 2009. 40 f. Tese (Graduação) – Universidade de Brasília, Brasília, 2009.
- Daronco, Leonardo Crauss, Avaliação Subjetiva de Qualidade Aplicada à Codificação de Vídeo Escalável, Dissertação para obtenção do grau de Mestre em Ciência da Computação, Porto Alegre, 2009.
- Duma, Luiz H.; Fonseca, Keiko V. O.; Pohl, Alexandre de A. P.. Técnicas de otimização para codificadores H.264. Curitiba, 2010.
- Gonzalez, Rafael C.; Woods, Richard E. Processamento de Imagens Digitais. São Paulo - Sp: Edgard Blücher Ltda., 2003.
- Greco, Felipe Silveira. Redes Neurais Para Reconhecimento de Padrões de Vídeo Mpeg4. 2011. 37 f. Trabalho de Conclusão de Curso de Engenharia Elétrica (3º Grau) - Universidade Federal do Paraná, Curitiba, 2011.
- Medeiros, Tiago Pizzetti. Segmentação Quadtree para Seleção de Blocos na Codificação de Vídeo H.264/AVC. 2011. 121 f. Trabalho de Conclusão de Curso (Bacharel) - Universidade do Extremo Sul Catarinense - Unesc, Criciúma, 2011.
- Porto, Marcelo Schiavon. Desenvolvimento Algorítmico e Arquitetural para Estimação de Movimento na Compressão de Vídeo de Alta Definição. 2012. 166 f. Tese (Doutorado) - Curso de Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2012.
- Rosa, Leandro Zanetti Paiva da. Investigação sobre Algoritmos para a Estimação de Movimento na Compressão de Vídeos Digitais de Alta Definição: Uma Análise Quantitativa. 2007. 155 f. Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Pelotas, Pelotas, 2007.
- Silva, André Marcelo Coelho da. Um Estudo Sobre o Padrão H.264/AVC de Compressão de Vídeo. 2007. 44 f. Dissertação (Pós-graduação) - Universidade Católica de Pelotas, Pelotas, 2007.
- Segall, C.A.; Sullivan, G.J. Spatial Scalability Within the H.264/AVC Scalable Video Coding Extension. IEEE Transactions on Circuits and Systems for Video Technology, Vol. 17, No. 9, pp. 1121-1135. Sept. 2007.
- Soler, Luciano. Compactação de Vídeo Escalável. 2006. 186 f. Dissertação (Mestrado) - Universidade Federal do Rio Grande do Sul, Porto Alegre, 2006.
- Souza, Diego Felix de. Codificação de Vídeo Via Casamento de Padrões Usando Estimação de Movimento. 2010. 88 f. Dissertação (Mestrado) - Curso de Programa de Pós-graduação em Engenharia Elétrica, Coppe, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2010.
- Wien, Mathias; Schwarz, Heiko; Oelbaum, Tobias, Performance Analysis of SVC. IEEE Transactions on Circuits and Systems for Video Technology, Vol. 17, No. 9, p. 1194-1203, Sept. 2007