

# Codificador de Vídeo *Full HD* em Tempo Real com Complexidade Escalável Usando a Biblioteca X264

Allan F. da Silva, José F. L. de Oliveira, Eduardo A. B. da Silva

**Resumo**—Este trabalho tem por objetivo a codificação de vídeo *Full HD* em tempo real, utilizando a biblioteca x264. Esta biblioteca implementa em *software* um codificador H.264/AVC de alto desempenho cujo conjunto de parâmetros de configuração contém um grande número de opções. Desta forma, é proposta uma metodologia para a escolha de um subconjunto relevante de parâmetros que permita a codificação em tempo real, com máxima qualidade de vídeo para uma dada complexidade computacional. Com base nos resultados obtidos utilizando esta metodologia, foi selecionado um conjunto de configurações que proporciona uma codificação em tempo real com a máxima qualidade para uma dada complexidade. Os resultados mostram que o sistema otimizado possui desempenho significativamente superior aos perfis predefinidos na biblioteca estudada. Em termos de PSNR, o aumento de desempenho é superior a 1 dB.

**Palavras-Chave**—Vídeo digital, H.264, x264, Compressão de vídeo.

**Abstract**—The objective of this work is to perform real-time full HD video encoding by using the x264 library. This library implements in software a high performance H.264/AVC encoder whose set of configuration parameters contains a large number of options. Thus, a methodology is proposed in order to select a relevant subset of parameters which allow real-time encoding and maximum video quality for a given computational complexity. Based on the results of this methodology, it was selected a set of settings that allow real-time encoding and maximum quality for a given complexity. The results show the optimized system has a significant performance improvement in relation to the predefined x264 profiles. In terms of PSNR, the improvement in performance is superior to 1 dB.

**Keywords**—Digital video, H.264, x264, Video compression.

## I. INTRODUÇÃO

O desenvolvimento tecnológico dos últimos anos proporcionou um avanço na compressão de vídeos digitais, permitindo sua ampla utilização em transmissões via internet ou aplicações em tecnologias móveis. Este avanço tecnológico culminou no surgimento do padrão de codificação de vídeo conhecido como H.264 [1], que possui excelente desempenho em termos de taxa-distorção, sendo hoje o padrão mais utilizado no mundo, estando presente inclusive no padrão de TV digital brasileiro.

O referido padrão apresenta diversos perfis de operação, de forma a abranger um grande número de aplicações, podendo ser usado em sistemas com variadas taxas de *bits* ou variados níveis de poder computacional, cada qual com uma certa eficiência. Devido a esse grande número de possibilidades, o processo de configuração de um codificador H.264 não é trivial. Mas, superada essa dificuldade da configuração, o H.264 permite reduzir a complexidade computacional de

codificação ao custo de eventuais perdas na qualidade do vídeo obtido.

Dentre as várias implementações de codificadores H.264 por *software* existentes, uma das principais é a do x264 [2], por ser de código aberto, apresentar alto desempenho em termos de velocidade de codificação [3] em comparação com *software* de referência [4], e por possuir uma grande quantidade de parâmetros, os quais permitem configurar melhor os diversos algoritmos implementados. Entretanto, este grande número de parâmetros também dificulta o seu uso. Neste trabalho, é apresentada então uma análise dos principais modos de operação desta biblioteca, visando estabelecer um conjunto de configurações que seja próximo do ótimo em termos de distorção *versus* complexidade para diversas taxas. Assim, será possível avaliar o uso deste codec para aplicações em tempo real com vídeos em resolução *full HD*, ou seja, de  $1920 \times 1080$  píxeis.

Diversos estudos propõem melhorias ao x264, valendo-se do fato de esta biblioteca possuir código aberto. Um exemplo está em [5], onde é proposta uma alteração na estrutura do programa de forma a reduzir o tempo gasto pelo computador com o acesso à memória. Este não é um dos objetivos contemplados por este trabalho, e sim gerar uma otimização em alto nível, a partir da escolha e configuração adequada dos parâmetros contidos no programa.

No trabalho desenvolvido em [6], é apresentada uma curva ótima de configurações, gerada a partir da variação conjunta de parâmetros selecionados *a priori*. Porém, no presente trabalho, também foi feita uma análise individual dos parâmetros, de forma a escolher quais devem ser fixados e quais devem ser combinados, ou seja, variados conjuntamente. Além disso, o enfoque do presente trabalho é gerar um conjunto de configurações que permita a codificação em tempo real de forma ótima, ao passo que em [6] o objetivo final é propor mudanças que melhorem a eficiência energética do codificador.

A seguir, na seção II, é apresentada a metodologia proposta para selecionar os parâmetros a serem fixados e os que serão variados conjuntamente. Na seção III, é feita uma descrição sucinta da base de dados de vídeos utilizada nos testes dos parâmetros. Na seção IV, são descritos os resultados experimentais e, por fim, as conclusões são relatadas na seção V.

## II. METODOLOGIA PROPOSTA

Como discutido anteriormente, o padrão H.264 possui diversas opções de atuação, que se refletem em uma grande quantidade de parâmetros a serem configurados, o que inviabiliza a simulação de todas as combinações possíveis entre os parâmetros. Dessa forma, é preciso selecionar inicialmente um subconjunto de parâmetros a ser testado, os valores que cada um irá assumir e como serão combinados durante as simulações com a biblioteca x264.

Para estabelecer esta condição inicial, foram realizados experimentos que envolvem uma análise particular de cada um dos parâmetros considerados mais relevantes ao sistema. Foram escolhidos aqueles que realizam o controle de diversas características da compressão de vídeos, com os principais sendo:

- **me** - o algoritmo de estimação de movimento;
- **subq** - o refinamento a nível de subpíxel para a estimação de movimento;
- **chroma\_me** - considera a informação de croma durante a busca na estimação de movimento a nível de subpíxel;
- **cabac** - o uso de codificação aritmética adaptativa;
- **deblock** - o filtro de desbloqueio;
- **8x8dct** - o tamanho da DCT (*Discrete Cosine Transform*);
- **partitions** - o tipo de partições dos blocos;
- **bframes** - a quantidade de quadros **B** consecutivos;
- **b\_adapt** - decidir automaticamente quando usar quadros **B** e sua quantidade até o máximo definido por **bframes**;
- **b\_pyramid** - permitir que quadros **B** sejam usados como referência para prever outros quadros;
- **mbtree** - controle de taxa por árvore de macroblocos;
- **rc\_lookahead** - ajusta a distância de *lookahead* da árvore de macroblocos;
- **trellis** - quantização ótima para taxa-distorção;
- **mixed\_refs** - permite que cada partição  $8 \times 8$ ,  $16 \times 8$  ou  $8 \times 16$  selecione de forma independente o seu quadro de referência;
- **weightp** - modo de predição ponderada para os quadros **P**;
- **weight\_b** - uso de predição ponderada em quadros **B**;
- **direct\_pred** - determina o tipo de predição de movimento usada para macroblocos diretos em quadros **B**;
- **ref** - o número de quadros de referência;
- **aq\_mode** - o modo de quantização adaptativa. Define como a quantização adaptativa (AQ) distribui *bits*.

Os testes foram analisados em comparação a uma configuração estabelecida como padrão, onde cada um dos parâmetros foi mantido em seu modo de operação menos complexo. Assim, a avaliação da velocidade foi feita em relação à maior velocidade obtida pelo codificador.

Variando cada parâmetro pré-selecionado, foram feitas medidas de **qualidade** e de **complexidade**. A qualidade foi medida em termos da relação sinal-ruído de pico, PSNR – *Peak Signal to Noise Ratio*, em dB. Já a complexidade foi medida em termos da velocidade em que o programa realizou a codificação, em quadros por segundo. Foram então determinados os parâmetros mais propícios a serem mantidos em um dado estado/valor, sem grande perda na otimização do sistema.

Após esta investigação, realizaram-se enfim simulações com as diversas combinações entre os parâmetros que não foram fixados, realizando as mesmas medidas mencionadas anteriormente, gerando uma nuvem de pontos no espaço qualidade (PSNR) *versus* complexidade (taxa de codificação). A maximização da qualidade obtível levou à determinação de um conjunto de configurações ótimas, que permitem realizar a codificação com a maior eficiência possível dada

uma complexidade requerida. Para validação, o resultado foi comparado às configurações padrão presentes na biblioteca, determinadas pelo parâmetro **preset**, com seus 4 modos menos complexos: **ultrafast**, **superfast**, **veryfast** e **faster**.

### III. BASE DE DADOS

A base de dados utilizada consiste de vídeos sem compressão em resolução *full HD* gravados no Laboratório de Processamento de Sinais (LPS) [7] e em uma biblioteca da TV Futura [8], além de algumas sequências de vídeo utilizadas na literatura [9]. Estas sequências foram combinadas, gerando 6 vídeos a serem usados nos experimentos. Cada um dos 4 primeiros vídeos é composto de sequências concatenadas, escolhidas arbitrariamente entre as sequências *crowd run*, *ducks take off*, *into tree*, *old town*, *park joy*, *blue sky*, *pedestrian area*, *riverbed*, *rush hour*, *station 2*, *sunflower* e *tractor*, de forma a se utilizar, no conjunto total de vídeos, uma ou duas vezes cada sequência. Para os vídeos 5 e 6, foram utilizadas, respectivamente, algumas das sequências gravadas na TV Futura e no LPS. Todos os vídeos foram gerados de forma a ter duração de um a dois minutos.

### IV. RESULTADOS EXPERIMENTAIS

Para cada simulação de determinado vídeo e configuração, foram efetuadas 30 execuções do programa, para gerar alguma informação estatística nos resultados obtidos. Além disso, o número de *threads* usados pelo programa durante a codificação foi reduzido a 1, para evitar que problemas relacionados ao paralelismo dos processos possam influenciar as medidas. Todos os testes foram executados em um computador com um processador Intel® Core™ i7 950 de 3,07 GHz, 8 GB de memória RAM e 2 discos rígidos de 1 TB e 7 200 rpm.

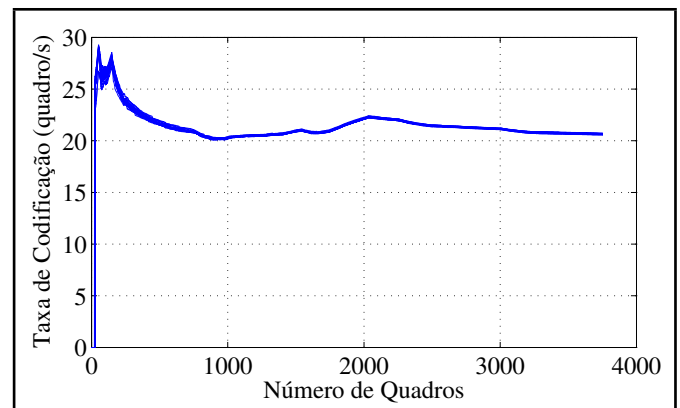


Fig. 1

GRÁFICO DA TAXAS DE QUADROS ATINGIDAS PELO CODIFICADOR EM FUNÇÃO DO NÚMERO DE QUADROS CODIFICADOS.

Um primeiro problema observado está retratado na Figura 1. Neste gráfico, que mostra diversas realizações sobrepostas de uma mesma simulação da taxa de codificação em função do número de quadros comprimidos, percebe-se um efeito indesejado de regime transitório do codificador para aproximadamente 800 quadros iniciais. Dessa forma, as simulações devem ser executadas preferencialmente em sequências de duração acima de 30 s, para que se obtenha o valor de velocidade em que o codificador se estabiliza.

### A. Análise Individual de Parâmetros

Foram efetuados testes para avaliação dos principais parâmetros da biblioteca estudada. Cada parâmetro teve seu desempenho analisado em relação a uma configuração de menor complexidade computacional. Para cada um, foi avaliado seu impacto em qualidade e complexidade, de forma a definir o melhor ponto de operação do mesmo.

As Tabelas I e II exibem o resultado obtido pelos algoritmos de estimação de movimento *diamond* (**dia**) [10], *hexagon* (**hex**) [11] e *uneven multi-hexagon* (**umh**) [12]. Os valores de qualidade e complexidade obtidos mostram que, em comparação ao algoritmo *diamond*, as técnicas mais complexas não trazem um ganho significativo de qualidade, tanto para o algoritmo *hexagon* quanto para o *uneven multi-hexagon*, ao passo que diminuem consideravelmente a velocidade de codificação.

TABELA I

MEDIDAS DE QUALIDADE E COMPLEXIDADE PARA O ALGORITMO DE ESTIMAÇÃO DE MOVIMENTOS, COM O VÍDEO 1 E TAXA DE BITS DE 10 Mb/s.

Algoritmo	dia	hex	umh
Qualidade	39,41	39,42	39,41
Taxa de Quadros	19,51	19,27	17,66
Desvio-padrão	0,28	0,13	0,15

Qualidade (PSNR) em dB. Taxa de quadros e desvio-padrão em quadro/s.

TABELA II

MEDIDAS DE QUALIDADE E COMPLEXIDADE PARA O ALGORITMO DE ESTIMAÇÃO DE MOVIMENTOS, COM O VÍDEO 2 E TAXA DE 10 Mb/s.

Algoritmo	dia	hex	umh
Qualidade	28,33	28,36	28,40
Taxa de Quadros	33,12	30,62	22,65
Desvio-padrão	1,27	1,27	0,40

Qualidade (PSNR) em dB. Taxa de quadros e desvio-padrão em quadro/s.

Outro teste, mostrado nas Tabelas III e IV, revela o impacto do aumento do número de quadros **B** consecutivos na codificação. Conforme se aumenta a quantidade de quadros **B**, a qualidade do vídeo aumenta, ao passo que a complexidade também aumenta. Porém, a partir de 3 quadros, a qualidade se mantém praticamente constante. Dessa forma, não é vantajoso elevar o número de quadros para além deste limite.

TABELA III

MEDIDAS DE QUALIDADE E COMPLEXIDADE PARA A MÁXIMA QUANTIDADE DE QUADROS B CONSECUTIVOS, COM O VÍDEO 1 E TAXA DE BITS DE 10 Mb/s.

Quadros B	0	1	2	3	16
Qualidade	39,41	39,92	39,98	40,04	40,04
Taxa de Quadros	19,51	18,44	18,23	18,07	18,03
Desvio-padrão	0,28	0,16	0,20	0,13	0,04

Qualidade (PSNR) em dB. Taxa de quadros e desvio-padrão em quadro/s.

Um exemplo mais distinto está retratado na Tabela V, que mostra os testes com algoritmos de estimação de movimentos a nível de subpíxel, indo desde o modo 0 com precisão de

TABELA IV

MEDIDAS DE QUALIDADE E COMPLEXIDADE PARA A MÁXIMA QUANTIDADE DE QUADROS B CONSECUTIVOS, COM O VÍDEO 2 E TAXA DE BITS DE 10 Mb/s.

Quadros B	0	1	2	3	16
Qualidade	28,33	29,03	29,06	29,09	29,09
Taxa de Quadros	33,12	30,68	29,50	29,38	29,05
Desvio-padrão	1,27	1,27	1,27	1,27	0,60

Qualidade (PSNR) em dB. Taxa de quadros e desvio-padrão em quadro/s.

píxel inteiro até o modo 3 com precisão de quarto de píxel e passo de meio píxel. A tabela mostra que a simples análise deste parâmetro não define a escolha do melhor modo de operação, já que ele permite uma certa troca entre qualidade e complexidade.

TABELA V

MEDIDAS DE QUALIDADE E COMPLEXIDADE PARA DIFERENTES MODOS DE PREDIÇÃO A NÍVEL DE SUBPÍXEL, COM O VÍDEO 3 E TAXA DE BITS DE 10 Mb/s.

Modo Subpíxel	0	1	2	3
Qualidade	38,40	38,88	39,06	39,15
Taxa de quadros	21,25	19,25	16,52	16,00
Desvio-padrão	0,27	0,15	0,60	0,30

Qualidade (PSNR) em dB. Taxa de quadros e desvio-padrão em quadro/s.

A Tabela VI mostra um resultado semelhante. Neste caso, está retratado o teste realizado com o parâmetro que rege o número de quadros anteriores que um quadro **P** pode usar como referência. Também para este parâmetro, somente os resultados apresentados não são capazes de definir um ponto ótimo de operação.

TABELA VI

MEDIDAS DE QUALIDADE E COMPLEXIDADE PARA DIFERENTES QUANTIDADES DE QUADROS ANTERIORES A SEREM USADOS COMO REFERÊNCIA PARA UM QUADRO P, COM O VÍDEO 2 E TAXA DE BITS DE 10 Mb/s.

Referências	1	2	3	4
Qualidade	28,33	28,41	28,44	28,47
Taxa de quadros	33,12	28,63	25,15	22,43
Desvio-padrão	1,27	0,44	0,25	0,17

Qualidade (PSNR) em dB. Taxa de quadros e desvio-padrão em quadro/s.

A análise descrita acima levou às escolhas para os parâmetros listados na Tabela VII. Foram definidos os melhores modos de operação para diversos parâmetros, e para dois deles, **subq** e **ref**, não foi possível definir a melhor escolha, de forma que eles deverão ser testados a partir de uma análise conjunta dos mesmos.

### B. Análise Combinada de Parâmetros

Feita a análise individual dos parâmetros e tendo sido tomada a decisão sobre quais deverão permanecer com valores fixos e quais serão variados de forma combinada, foram realizados testes com as combinações entre os que serão variados, **subq** e **ref**, assumindo os valores listados na Tabela VIII.

TABELA VII  
MODOS ÓTIMOS DE OPERAÇÃO PARA OS PARÂMETROS TESTADOS INDIVIDUALMENTE.

Parâmetro	Configuração	Parâmetro	Configuração
8x8dct	1	trellis	1
me	dia	deblock	1
chroma_me	0	bframes	3
b_adapt	1	rc_lookahead	0
b_pyramid	1	partitions	i8x8, i4x4
mbtree	0	weight_b	0
mixed_refs	0	direct_pred	spatial
weightp	0	ref	não definido
aq_mode	0	subq	não definido
cabac	1	***	***

Não foram considerados os modos mais complexos, porque as simulações demandariam um tempo muito maior, e os resultados fugiriam do escopo de codificação em tempo real. Desse modo, foi encontrada uma curva com os parâmetros ótimos para diferentes velocidades e taxas de *bits*. Exemplos desta curva com sua respectiva nuvem de pontos são apresentados nas Figuras 2 e 3, para o vídeo 2, e Figura 4 para o vídeo 6, com diferentes taxas de *bits*. Para análise dos resultados, foi realizada a comparação com padrão do x264, como mostrado nas Figuras 5, 6 e 7.

TABELA VIII  
PARÂMETROS TESTADOS COMBINADAMENTE.

Parâmetro	Valores Assumidos
subq	modo 0 até modo 4
ref	1 a 4 quadros de referência

Em qualquer um dos gráficos das Figuras 5, 6 e 7, percebe-se que as curvas ótimas apresentadas correspondem a uma faixa mais estreita de valores possíveis de complexidade, devido a poucos parâmetros terem sido combinados. Porém, para estas curvas, o desempenho em relação à qualidade foi até 1,3 dB superior ao obtido pelo padrão. Além disso, devido à derivada das curvas, para a curva otimizada a qualidade do vídeo gerado pelo codificador apresenta menor sensibilidade em relação à complexidade, o que é vantajoso, pois, ao utilizar as configurações ótimas, caso seja necessário um aumento na velocidade de execução do programa, o impacto na qualidade obtida será reduzido.

### C. Codificação em Tempo Real

A análise apresentada foi realizada com uma limitação no número de processos simultâneos (*threads*), porque as diferentes configurações não apresentam o mesmo desempenho em relação ao multiprocessamento. Assim, sob as mesmas condições, as comparações foram realizadas com maior acurácia. Já a Figura 8, em contraste com a Figura 6, mostra o contrário: a partir do sistema otimizado, a codificação foi executada de forma a aproveitar os vários núcleos do processador. Com 4 *threads*, o sistema se mostrou capaz de codificar o vídeo 2 para uma taxa de *bits* alvo de 10 Mb/s com um desempenho novamente superior ao padrão, mas atingindo velocidade de

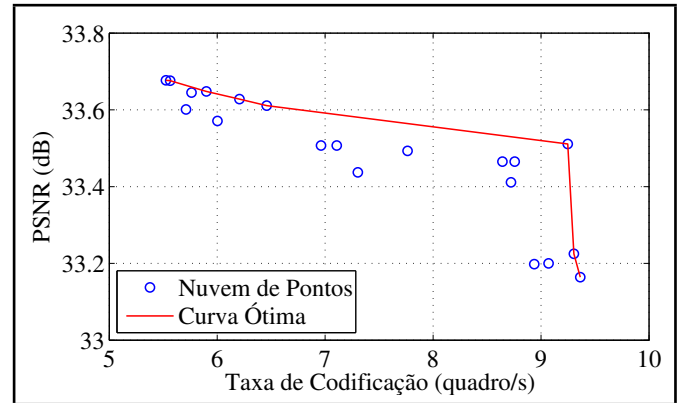


Fig. 2

CURVA ÓTIMA PARA O CONJUNTO DE CONFIGURAÇÕES DO VÍDEO 2, COM TAXA DE BITS DE 1 Mb/S.

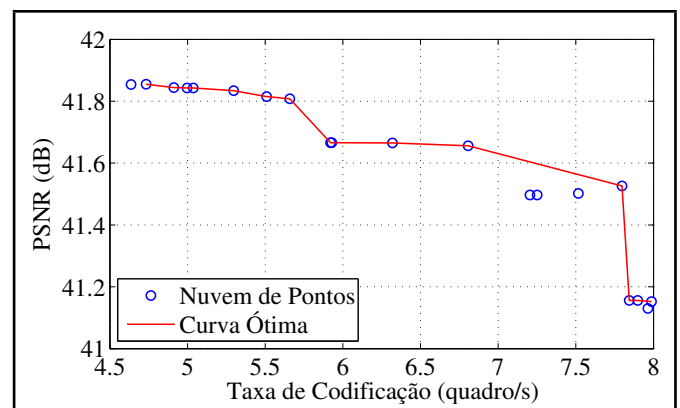


Fig. 3

CURVA ÓTIMA PARA O CONJUNTO DE CONFIGURAÇÕES DO VÍDEO 2, COM TAXA DE BITS DE 10 Mb/S.

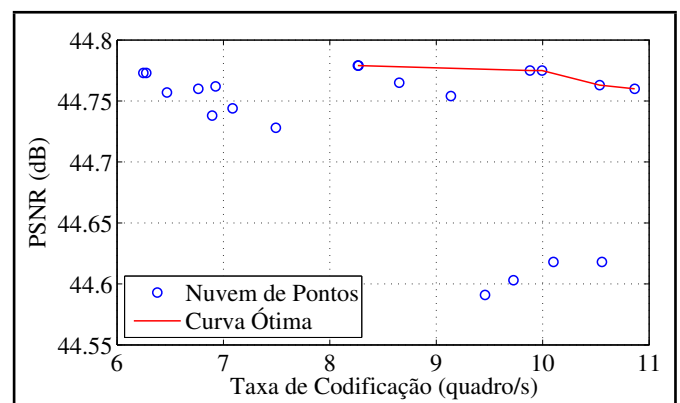


Fig. 4

CURVA ÓTIMA PARA O CONJUNTO DE CONFIGURAÇÕES DO VÍDEO 6, COM TAXA DE BITS DE 2 Mb/S.

até 28 quadros por segundo, possibilitando assim sua aplicação em tempo real.

## V. CONCLUSÕES

Neste artigo, estuda-se a codificação de vídeo *full HD* em tempo real através do codificador H.264/AVC implementado em *software* na biblioteca x264 [2]. Com esta finalidade, o impacto dos parâmetros de configuração na

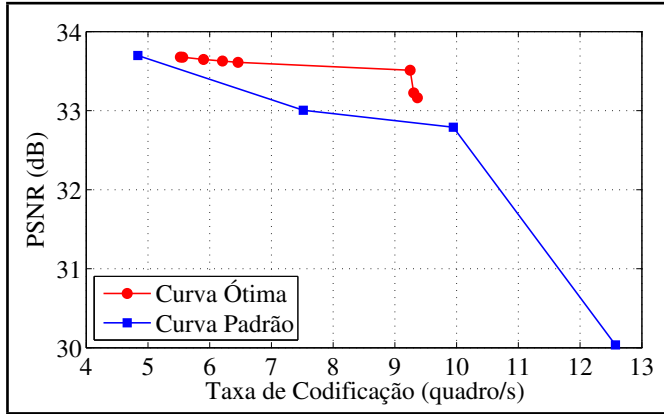


Fig. 5

COMPARAÇÃO ENTRE A CURVA ÓTIMA E A CURVA GERADA PELO PADRÃO DO X264 PARA O VÍDEO 2, COM TAXA DE BITS DE 1 Mb/s.

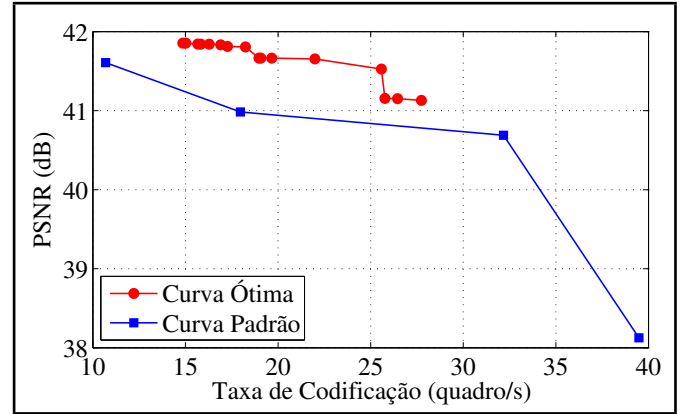


Fig. 8

CURVA ÓTIMA PARA O CONJUNTO DE CONFIGURAÇÕES DO VÍDEO 2, COM TAXA DE BITS DE 10 Mb/s E 4 THREADS.

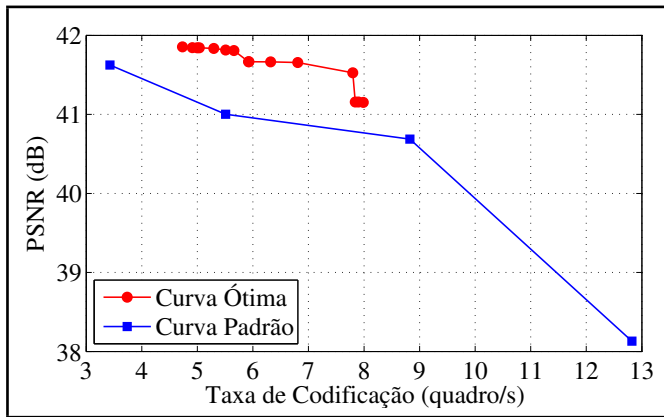


Fig. 6

COMPARAÇÃO ENTRE A CURVA ÓTIMA E A CURVA GERADA PELO PADRÃO DO X264 PARA O VÍDEO 2, COM TAXA DE BITS DE 10 Mb/s.

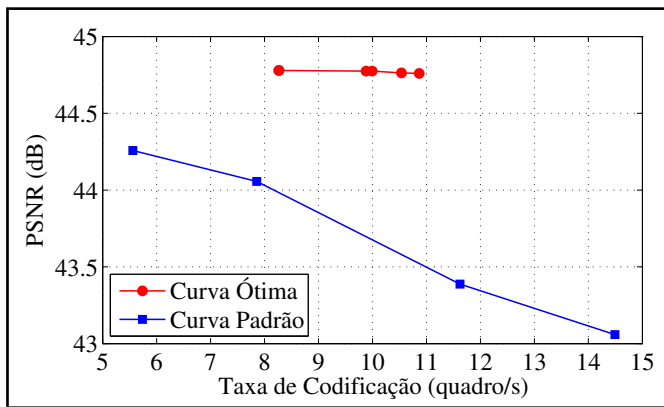


Fig. 7

COMPARAÇÃO ENTRE A CURVA ÓTIMA E A CURVA GERADA PELO PADRÃO DO X264 PARA O VÍDEO 6, COM TAXA DE BITS DE 2 Mb/s.

qualidade/complexidade da codificação de vídeo foi avaliado através de uma série de experimentos. A realização dos experimentos consistiu de diversas simulações utilizando a biblioteca x264 para um conjunto de configurações, onde foram medidas a qualidade do vídeo obtido, em termos de PSNR, e a respectiva complexidade em termos de velocidade de codificação, em quadros por segundo. A análise dos

resultados permitiu a determinação dos melhores modos de atuação do codificador. Os resultados experimentais também demonstraram que estas configurações obtidas apresentam desempenho superior aos perfis de configuração presentes na implementação da biblioteca estudada. Em termos de PSNR, o aumento de desempenho chega a ser superior a 1 dB em diversos casos. Dessa forma, a codificação de vídeos pode ser realizada de forma a obter a máxima qualidade para uma mesma complexidade computacional. Por fim, foi verificado que o sistema otimizado é capaz de realizar a codificação com velocidade suficiente para aplicações em tempo real.

REFERÊNCIAS

- [1] JVT of ISO/IEC MPEG, ITU-T VCEG, *Advanced Video Coding for Generic Audiovisual Services*, ITU-T Recommendation H.264 03/2005, ITU - International Telecommunication Union, Março 2005.
- [2] X264, "Biblioteca X264 da VideoLAN.org", <http://www.videolan.org/developers/x264.html>, Acessado em Maio de 2013.
- [3] MERRITT, L., VANAM, R., "Improved Rate Control and Motion Estimation for H.264 Encoder". *IEEE International Conference on Image Processing*, San Antonio, Texas, Setembro-Outubro 2007.
- [4] SOFTWARE JM, "Software do Codificador de Referência H.264/AVC", <http://iphome.hhi.de/suehring/tml>, Acessado em Maio de 2013.
- [5] DUMA, L. H., *Uma Proposta de Método para Melhoria de Desempenho do Codificador x264 Baseada na Análise do Acesso ao Barramento Externo de Memória*. Dissertação de Mestrado, Engenharia Elétrica e Informática Industrial, Universidade Tecnológica Federal do Paraná, Curitiba, 2011, Agosto 2011.
- [6] FONSECA, T. A., *Codificação de Vídeo Escalonável em Complexidade e em Energia*. Tese de Doutorado, Faculdade de Tecnologia, Departamento de Engenharia Elétrica, Universidade de Brasília, Agosto 2012.
- [7] LPS, "Sequências full HD gravadas no Laboratório de Processamento de Sinais", <http://www.lps.ufrj.br/~tvdigital/muque/data/lps/original>, Acessado em Junho de 2012.
- [8] FUTURA, "Sequências full HD gravadas na biblioteca da TV Futura", <http://www.lps.ufrj.br/~tvdigital/muque/data/futura/original>, Acessado em Junho de 2012.
- [9] XIPH, "Sequências full HD da Xiph.org Video Test Media", <http://media.xiph.org/video/derf>, Acessado em Maio de 2013.
- [10] ZHU, S., MA, K.-K., "A New Diamond Search Algorithm for Fast Block-matching Motion Estimation", *IEEE Transactions on Image Processing*, v. 9, n. 2, pp. 287–290, Fevereiro 2000.
- [11] ZHU, C., LIN, X., CHAU, L.-P., "Hexagon-Based Search Pattern for Fast Block Motion Estimation", *IEEE Transactions on Circuits and Systems for Video Technology*, v. 12, n. 5, pp. 349–355, Maio 2002.
- [12] XU, X., HE, Y., "Improvements on Fast Motion Estimation Strategy for H.264/AVC", *IEEE Transactions on Circuits and Systems for Video Technology*, v. 18, n. 3, pp. 285–293, Março 2008.