

Codificação de Vídeo H.264/AVC em Tempo Real e Restrita por Energia

Tiago A. da Fonseca* e Ricardo L. de Queiroz⁺

Resumo—Comunicações de vídeo têm demandado poder de computação cada vez maior e, portanto, maior consumo de energia. Para permitir soluções *verdes* de codificação de vídeo baseadas em *software*, propomos um método para otimizar o estágio de predição do codificador H.264/AVC em termos de energia (E) e o testamos em uma implementação em *software*. Além de buscar as melhores opções de codificação para o melhor desempenho em termos de RD (*Rate-distortion*), o processo é restrito para adequar-se a uma determinada provisão de energia por meio de otimização no espaço RDE . Nós medimos E pela energia elétrica demandada por um determinado sistema de codificação. Nosso esquema otimizado por RDE permite compressão de vídeo por *software* em tempo real restrita a um alvo de consumo de energia. Nós mostramos resultados para compressão restrita por energia e, em alguns casos, é possível economizar até 35% da energia com impacto mínimo no desempenho RD .

Palavras-Chave—H.264, predições, escalabilidade de energia, controle de taxa, codificação de vídeo em tempo real.

Abstract—Video communications have demanded higher computing power and, therefore, higher energy expenditure. To provide greener software-based video encoding solutions, we studied a method to optimize the prediction stage of H.264/AVC in the energy sense (E) and tested it with a software implementation. Besides looking for the coding options to the best RD -performance, the process is constrained to fit within a certain energy budget by an RDE optimization. We measure E as the real demanded electric energy for a given system. Our RDE -optimized framework allows for software-based real-time video compression constrained electrical consumption. We show results of energy-constrained compression and in some cases one can save as much as 35% of the energy with small impact on RD performance.

Keywords—H.264, predictions, energy scalability, rate control, real-time video coding.

I. INTRODUÇÃO

Historicamente, fabricantes de processadores responderam pela demanda de maior poder de processamento basicamente com processadores mais velozes, com maior frequência de relógio. Contudo, maiores frequências de relógio implicam e maiores consumo de potência e dissipação térmica. Atualmente, o uso de energia e as emissões de carbono são itens de grande relevância. Pessoas, empresas e outras organizações estão em busca de produtos eficientes energeticamente pois os custos com energia elétrica vêm crescendo e tornando-se um elemento importante na composição de custos operacionais [4]. Assim, observa-se que baixo consumo de energia tem se tornado uma das mais importantes restrições de desempenho no projeto de dispositivos eletrônicos. A indústria, por sua vez,

tem oferecido novas arquiteturas com forte viés de economia de energia além de algoritmos modernos mais eficazes.

Processamento de imagem e vídeo são componentes importantes por trás dessa busca por poder de computação e um dos mais modernos padrão de compressão de vídeo, o H.264/AVC [1]-[3], é um exemplo de aplicação de elevada carga computacional. Desenvolvido segundo a arquitetura de codificador DPCM híbrido [5], o H.264/AVC busca pela melhor predição do sinal que está sendo codificado de forma a gerar a representação mais compacta [6]. Todavia, o processo de busca do predições é uma das etapas mais demoradas de um codificador de vídeo baseado em *software* [7]. Devido ao fato de os compressores de vídeo gastarem a maior parte do tempo de operação realizando predições, o controle da complexidade do estágio de predição surge como um dos primeiros itens para o controle da complexidade global do sistema e o consumo de energia. Além disso, ao se codificar sequências de vídeo em aplicações que exigem baixa latência, como vídeo-conferências, o tempo gasto na compressão do sinal é relevante e a codificação de vídeo em tempo real pode ser tecnicamente desafiadora.

Há vários trabalhos com foco na redução da complexidade do H.264/AVC. Alguns modificam o estágio de predições para reduzir a carga de computações com pequenas penalizações no desempenho RD [8]-[10], mas não tratam do consumo energético. No trabalho [11], o consumo de energia a tratado pelo ajuste de parâmetros específicos disponibilizados pelo processador que executou os testes. Um trabalho recente [12] apresenta resultados significativos para redução de complexidade do codificador H.264/AVC em sua implementação de referência, entretanto grande parte da escala em complexidade não será alcançada se o esquema for implementado usando algoritmos rápidos, bibliotecas de primitivas de alto desempenho [14] e novos recursos computacionais [13], que, por sua vez, podem demandar bastante energia.

Este trabalho sugere novas estratégias para a economia de energia. Propomos avaliar a atividade de compressão de vídeo por meio da análise das demandas de energia. Uma estratégia de otimização de taxa-distorção-energia (RDE) é apresentada como forma de restringir as demandas de energia do codificador num cenário de compressão de vídeo em tempo real. Em seguida, os parâmetros otimizados são usados para avaliar um esquema de codificação de vídeo em tempo real otimizado em termos de RDE . Inserimos modificações em aspectos não normativos do estágio de predição do codificador H.264/AVC para realizar codificação por *software* escalonável em energia com controle de taxa de transmissão e em tempo real. A abordagem proposta pode ser convenientemente ado-

*Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília-DF, Brasil, E-mail: tiago@image.unb.br

⁺Departamento de Ciência da Computação, Universidade de Brasília, Brasília-DF, Brasil, E-mail: queiroz@ieee.org

tada em sistema de comunicações móveis, cujo poder de computação tem crescido ao nível daquilo que é oferecido por um computador pessoal, haja vista que eficiência energética ainda é um problema para plataformas de computação restritas energeticamente (alimentadas por baterias, por exemplo) [15].

II. OTIMIZAÇÃO BASEADA EM ENERGIA

É comum que atividades de otimização tratem funções de custo ou taxas de sucesso. Tomemos um codificador em *software* que execute uma atividade para a qual nós desejamos, de alguma maneira, medir seu custo. Em compressão de sinais, costuma-se medir custos com medidas de qualidade, como a distorção (D), com medidas de taxa de bits (R) ou uma combinação de ambas. Assume-se o processo de compressão como uma atividade parametrizada, i.e., é possível que escolhamos os valores dos N parâmetros $\{P_i\}_{i=1,\dots,N}$. Tomemos \mathbf{P} como o vetor com todos os P_i . O codificador é executado sobre determinado conjunto Z de dados, que pode ser diferente a cada execução. Para cada escolha de \mathbf{P} e Z , nós podemos ter uma medida C do custo do codificador. Basicamente, nós podemos ter o seguinte mapeamento

$$C = f(\mathbf{P}, Z).$$

Outro atributo que pode ser derivado a partir de cada execução é o esforço envolvido na execução da própria codificação, que pode ser medido como a energia demandada $E = g(\mathbf{P}, Z)$. Espera-se que alguns parâmetros como número de iterações, tamanho do conjunto de dados etc influenciem a energia demandada enquanto outros não. A ideia central neste trabalho vem do fato de a correlação entre E e C ser afetada diferentemente ao se ajustar diferentes parâmetros. Usaremos essa propriedade para encontrar pontos que minimizam o consumo energético. A ideia é mostrada na Fig. 1, que apresenta uma nuvem de pontos, no espaço custo-energia, de todos os possíveis \mathbf{P} para um determinado sistema e determinados dados de entrada. A figura enfatiza um conjunto de pontos encontrados após a otimização, o casco convexo inferior (LCH, do inglês *Lower Convex Hull*) de todos os pontos, representados por quadrados. Pontos que pertencem ao LCH representam codificações em que o sinal é comprimido usando a menor energia para determinado custo, e é justamente a região em que deseja operar. Outro subconjunto discriminado é composto por pontos que representam a varredura nos valores de determinado parâmetro P_i , mantendo-se todos os outros fixos, e é representado por estrelas vermelhas. A varredura no intervalo de definição de um parâmetro pode resultar um conjunto subótimo, diferente do LCH. Uma dentre muitas formas de definir o LCH, uma solução simples, que leva a um conjunto marginalmente não convexo, consiste em incluir um ponto no LCH somente se nenhum outro ponto possuir, simultaneamente, valor de C inferior e valor de E inferior ao seu.

Apesar da explicação simples por meio do uso de um custo como variável escalar, em codificação de vídeo, o mapeamento é convenientemente tratado por meio de uma variável multidimensional $\mathbf{C} = [R, D]$, onde R representa a taxa de bits e D , a distorção (por exemplo, o erro quadrático médio). Desta forma, $C = f(\mathbf{P}, Z)$.

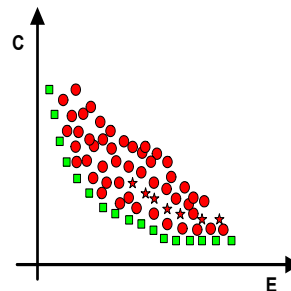


Fig. 1. Nuvem de pontos no espaço energia vs. custo. Os pontos do LCH são indicados por quadrados verdes. Um conjunto subótimo determinado, por exemplo, pela variação de apenas um dos parâmetros é ilustrado pelas estrelas vermelhas.

III. OTIMIZAÇÃO DE FUNÇÕES DE CUSTO RDE

\mathbf{P} e Z são mapeados para R , D and E , o que resulta na adição da dimensão da energia (E) ao problema de otimização por taxa e distorção¹. Desejamos encontrar os parâmetros que nos permitam operar no LCH do espaço RDE . Desta forma, deve-se garantir que nenhuma outra configuração resulte em menor consumo energético para dado valor de custo. Por outro lado, garante-se que, para um determinado patamar de consumo energético, nenhuma outra configuração do codificador resultará em melhor desempenho. A Fig. 2 ilustra o LCH no espaço RDE .

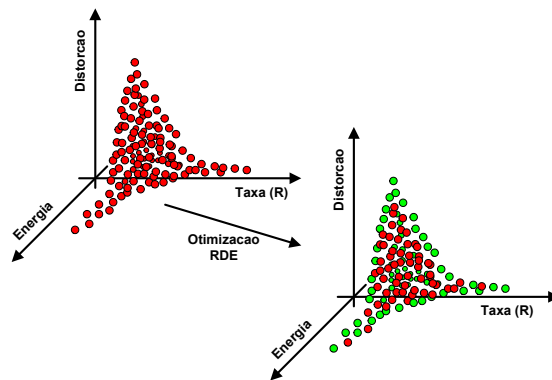


Fig. 2. Ilustração do conjunto de pontos RDE que constituem a frente de Pareto. Os pontos verdes fazem parte do casco convexo inferior (LCH). Note que alguns pontos estão oclusos devido ao ponto de vista.

Uma abordagem consiste em usar conjuntos de dados para treinamento. Tomemos $\{\mathbf{P}_k\}$ como o conjunto de todas as escolhas dos parâmetros, ordenados de modo conveniente. Considere, também, que \mathbf{P}_k tenha elementos P_{kn} . Se usarmos um conjunto de dados representativo \hat{Z} , é possível varrer $\{\mathbf{P}_k\}$, calcular E , R e D e identificar os pontos que pertencem ao LCH do espaço $E \times R \times D$. Para o n -ésimo ponto pertencente ao LCH, registramos $\mathbf{Q}_n = [E_n, R_n, D_n, \mathbf{P}_n]$, que contém os pontos ótimos para o conjunto representativo \hat{Z} , que podem ser considerados bons o suficiente para outros dados. O algoritmo para treinamento *off-line* é:

- 1) Apresente um conjunto de dados representativos \hat{Z} e crie uma lista vazia Q .

¹Medimos a potência ativa (em watts, W) a partir das leituras fornecidas por um wattímetro, das quais nós podemos derivar o consumo energético (em joules, J).

- 2) Para todos os k , calcule $E_k = g(\mathbf{P}_k, \hat{Z})$ e $[R_k, D_k] = f(\mathbf{P}_k, \hat{Z})$. Se o ponto pertencer ao LCH, insira $\mathbf{Q}_k = [E_k, R_k, D_k, \mathbf{P}_k]$ na lista Q .
- 3) Devolva a lista Q de pontos no LCH.

Depois de encontrar N_q pontos que pertençam ao LCH, ordena-se Q em ordem crescente de energia, i.e. E_l em Q é não decrescente. Ao se executar *on-line*, o algoritmo de busca de parâmetro é o seguinte: Inicie considerando a taxa de bits R^r (restrição do canal de transmissão) e o valor desejado de provisão de energia E^r . Então:

- 1) Apresente uma lista Q de pontos no LCH, o valor desejado de provisão de energia E^r e a taxa de bits alvo R^r . Crie uma lista vazia L .
- 2) Varra Q , para $k = 1, \dots, N_q$. Se $|R_k - R^r| < \epsilon$, insira \mathbf{Q}_k na lista L .
- 3) Conte N_l , o número de itens em L . Observe que os itens da lista L estão ainda em ordem crescente de energia e assuma que todos os parâmetros resultem em taxa de bits muito próximas.
- 4) Varra a lista L , para $k = 1, \dots, N_l$, até que $E_k \leq E^r \leq E_{k+1}$, então pare.
- 5) Encontre \mathbf{P}' por interpolação de \mathbf{P}_k e \mathbf{P}_{k+1} da lista L .
- 6) Devolva o vetor de parâmetros \mathbf{P}' .

O conjunto de parâmetros \mathbf{P}' passa a ser utilizado para comprimir o conjunto de dados Z . Utilizamos alvos energéticos E^r restritos a uma taxa de bits R^r , mas é trivial substituir a taxa alvo por um alvo de distorção D^r . Naturalmente, muitos parâmetros não assumem valores contínuos e alguns artifícios tem de ser usado para ajustá-los corretamente. Por exemplo, tome o caso para o m -ésimo parâmetro, em que é possível usar o valor P_{km} se $E^r - E_k < E_{k+1} - E^r$, caso contrário, usar o valor de $P_{k+1,m}$.

Se uma arquitetura de malha fechada for permitida, é possível monitorar o consumo de energia do sistema e ajustar, continuamente, os parâmetros. Discrepâncias entre Z e \hat{Z} podem provocar um descasamento entre o consumo energético observado e o previsto pelo modelo. Nesse caso, observa-se que \hat{Z} não é tão representativo como foi assumido inicialmente. Esse descasamento pode ser devidor a não linearidades do mapeamento g . Uma solução é iniciar com uma provisão alvo E^r e medir, periodicamente, a energia $E(n)$. Em seguida, adaptam-se os parâmetros de forma a controlar o gasto energético (ou custo). Assumamos que, em determinado instante n , \mathbf{P}' é tomado como uma interpolação de \mathbf{P}_j and \mathbf{P}_{j+1} . Se $E(n) < E^r$, é necessário mover \mathbf{P}' na direção de \mathbf{P}_{j+1} ou mesmo \mathbf{P}_{j+2} . Por outro lado, se $E(n) > E^r$, deve-se mover na direção oposta, i.e. para \mathbf{P}_j ou até mesmo para \mathbf{P}_{j-1} .

O laço de controle se beneficia de todas as propriedades de simples sistemas adaptativos e há muitas técnicas para escolher os mecanismos de adaptação e tratar de problemas de convergência. Nas próximas seções nós mostraremos exemplos da aplicabilidade do método proposto.

IV. OTIMIZAÇÃO RDE DO CODIFICADOR H.264/AVC

Nossa estratégia para a construção de um codificador de vídeo em tempo real e restrito por energia consiste no controle

da demanda de energia durante a codificação de uma sequência de vídeo por meio do ajuste de parâmetros do codificador de forma que as perdas de desempenho em termos de RD , em comparação com o máximo desempenho alcançável, sejam as menores possíveis. A energia (E) é medida pelo tratamento das leituras de potência realizadas por um wattímetro conectado ao computador responsável pela execução do compressor de vídeo em *software*. Optou-se pela implementação x264 do padrão H.264/AVC devido a sua velocidade de codificação e bom desempenho em termos de RD [16]. Aplicando a metodologia discutida na Seção III, nossa abordagem estende o conceito de otimização RD (*Rate-Distortion Optimization, RDO*) [17], com a adição da dimensão E .

O vetor \mathbf{P} é composto pelos seguintes parâmetros relacionados a esforço computacional: quantidade de quadros B ($\#B$), quantidade de quadros de referência ($\#Refs$), precisão dos vetores de movimentos (MVP) usados na compensação de movimentos, a seleção da heurística de decisão de modos de predição (MD), parâmetro de quantização (QP) e o número de *threads* usadas na compressão ($\#Thrds$):

$$\mathbf{P} = \{\#B, \#Refs, MVP, MD, QP, \#Thrds\}.$$

O custo \mathbf{C} é o vetor $[R, D]$, em que a taxa é medida em bits por segundo enquanto a distorção é calculada como o erro quadrático médio entre o sinal original e sua versão decodificada.

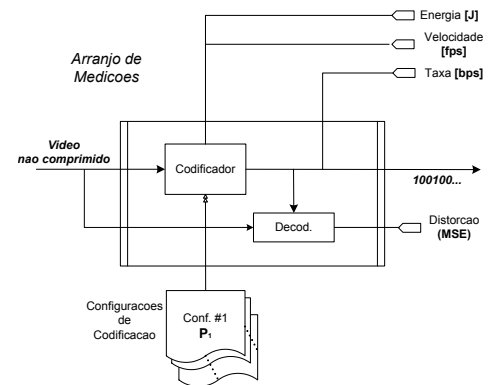


Fig. 3. Arranjo de medições: para cada execução do codificador (\mathbf{P}_k), os custos envolvidos ($\mathbf{C} = [R, D]$), a energia (E) e a velocidade de compressão são registrados. A partir desses pontos, o LCH de pontos no espaço RDE é determinado.

Escolheu-se avaliar uma aproximação empírica para a função de energia $E = g(\mathbf{P}, Z)$ por meio de medições (Fig. 3). Portanto, para cada ajuste de configuração do compressor de vídeo, calculamos a taxa de bits total (R), o MSE (D) e a energia (E) gasta na codificação de uma sequência de treinamento representativa. Os pontos RDE são posteriormente empregados na montagem do espaço de busca do qual será determinado o LCH por meio de otimização RDE . Depois de encontrar as configurações que levam o compressor a operar no LCH, constrói-se uma tabela de consultas para faixas operacionais de desempenho de forma a determinar os pontos RDE ótimos de partida para compressão no LCH. Pontos intermediários de energia não listados diretamente na tabela de consultas podem ser facilmente derivados por meio de interpolação de forma que a energia demandada seja muito próxima da provisão inicial, sem se afastar do LCH.

V. RESULTADOS

O método proposto foi implementado usando a implementação em *software* x264, executada num computador pessoal com o processador Intel® Core™ i7 PC (sem monitor de vídeo), com o codec executando na mais alta prioridade de escalonamento. A ênfase foi em sequências de vídeo 720p (1280×720-pixels) e SD (704×576-pixels); portanto, fez-se necessário utilizar estimação de movimento hexagonal (UmHexS) pelas restrições de tempo real. Para montar sequências representativas 720p, restringimos o escopo e optamos por tratar de sequências de vídeoconferência gravadas a 30 Hz. Para o conjunto de treinamento, usamos as sequências “Seq05”, “Seq06” e “Seq17”². As sequências SD padrões (60 Hz) “Harbour”, “Crew” e “Soccer” foram misturadas para montar um conjunto de quadros representativo.

A Figura 4(a) apresenta as curvas ótimas em termos de RD para a sequência de treinamento 720p, cada curva representando uma demanda energética diferente conforme indicado nas legendas (valores médios de potência): quanto mais elevada a demanda energética, melhor o desempenho em termos de RD alcançado pelo compressor. Na Figura 4(b), mostramos como as demandas energéticas variam à medida que se varia a taxa de bits segundo a qual se comprime o sinal. Taxas maiores demandam mais energia na codificação em tempo real de sequências 720p pois aumenta-se esforço necessário para quantizar precisamente os coeficientes transformados e, além disso, são mais numerosos os coeficientes para tratamento por codificação entrópica. Deve ser frisado que, em todos os pontos da curva, o codificador está trabalhando pelo menos na velocidade em que as sequências de vídeo foram capturadas (30 Hz ou 30 fps, *frames per seconds*): codificação em tempo real. Sempre que a velocidade de codificação for maior que a taxa com que os quadros chegam para codificação, o compressor deverá esperar “dormindo” enquanto novos quadros são disponibilizados.

Para analisar, de forma mais intuitiva, como as curvas RD de outras sequências se comportam com a variação das provisões de energia, optou-se por calcular diferenças de desempenho pelo cálculo das diferenças médias de PSNR entre duas curvas RD , conforme descrito na recomendação [18]. A referência de desempenho, em relação à qual as diferenças serão calculadas, é a configuração de parâmetros que permite o melhor desempenho em termos de RD , mas que permite a codificação em tempo real; desta forma, é a configuração (em tempo real) que demanda a maior quantidade de energia. Em termos relativos, essa referência de desempenho representará a razão de energia de 1,0J/J. Os resultados, em termos de média, são mostrados na Fig. 5(a) para sequências de vídeo 720p. O comportamento geral sugere que, na medida em que se reduz a quantidade de energia disponível, as penalidades em desempenho aumentam. Na Fig. 5(b), são apresentados resultados para sequências SD.

O resultado principal é um esquema de compressão de

²“Seq05” é uma cena em que há uma interlocutora sentada numa mesa (fundo de cena suave), por sua vez, na “Seq06”, há um interlocutor, o fundo de cena possui mais detalhado (alta frequência). A “Seq17”, além de um fundo detalhado, verifica-se a chegada de um segundo interlocutor na cena, o que muda os padrões de movimento.

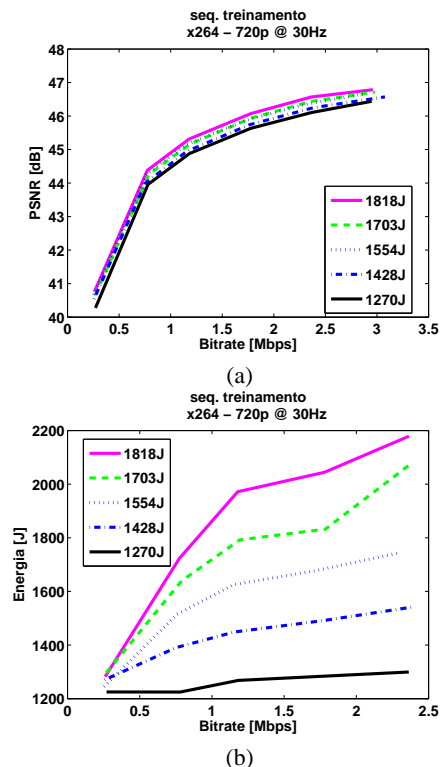


Fig. 4. Desempenho RD para diferentes perfis energéticos usando a sequência de treinamento (a) e o efeito da variação da taxa de bits na demanda de energia (b). Na figura (b), os valores da legenda representam o valor médio da energia demandada para cada uma das curvas da figura (a).

vídeo em tempo real e restrito por energia que permite o usuário escolher a provisão de energia segundo a qual deverá ser comprimida uma sequência de vídeo. Como esperado, o desempenho em termos de RD é penalizado na medida em que as provisões de energia são diminuídas. Contudo, as curvas são muito próximas umas das outras e o pior caso é representado por curvas de cenas com muitos objetos em movimento ou com alto detalhamento (texturas e outras informações de alta frequência). Para sequências que demandam menor esforço de codificador (“Seq 15” e “Seq 21”)³, o esquema proposto penaliza não mais que 1,3 dB em PSNR em média, enquanto resulta em 35% de economia de energia. Para as sequências SD, apesar de a velocidade mínima de compressão ser maior (60 fps), as perdas em PSNR foram menores (abaixo de 0,6 dB) para valores similares de economia de energia, mesmo para sequências com alto nível de detalhes. O ajuste de conjuntos de treinamento pode ainda melhorar esses números.

VI. CONCLUSÕES

Nós propusemos um esquema otimizado por energia para a codificação de vídeo em *software* pelo padrão H.264/AVC, esquema esse que permite codificação de vídeo em tempo real. Em vez de empregarmos todas as ferramentas disponíveis para alto desempenho de previsões de sinal, escolheu-se, de forma otimizada, apenas um subconjunto dessas ferramentas. Inicialmente, treinou-se o sistema e ajustaram-se de parâmetros para se obter o melhor desempenho RD restrito por uma determinada provisão de energia. Inseriu-se um módulo de controle

³Seq 15 e Seq 21 são cenas em que há um casal de interlocutores: o fundo da sequência 21 é mais detalhado que o fundo da sequência 15. Ambas foram gravadas a 30 Hz e comprimidas a 30 fps.

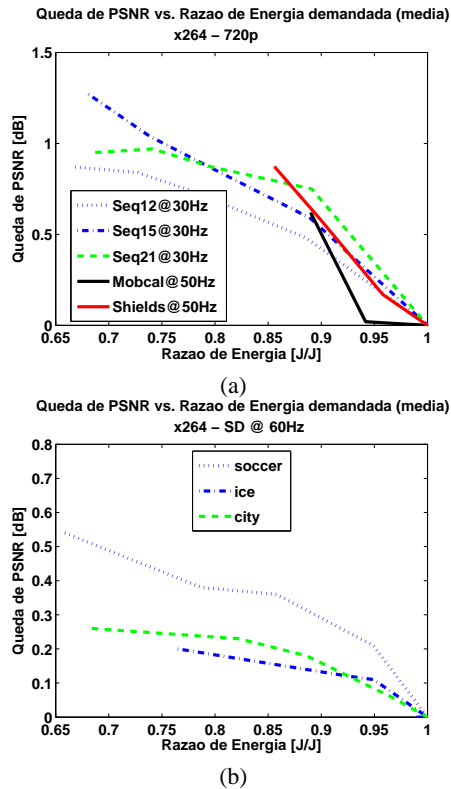


Fig. 5. Queda em PSNR vs. razão de energia consumida para seqüências (a) 720p e (b) SD. A qualidade do vídeo comprimido tende a crescer à medida que se aumentam as provisões de energia. A razão de energia de 1,0J/J corresponde ao caso do melhor desempenho *RD* alcançado em codificação em tempo real.

capaz de ajustar continuamente a velocidade de compressão e adequar a demanda energética. Nossos testes mostraram que o desempenho *RD* foi levemente afetado pelo esquema de restrição energética, que, a propósito, não usa descarte de quadros nem mudança de resolução para proporcionar escala em energia. Mesmo assim, ele é capaz de promover uma escala considerável de complexidade e energia. Em essência, o esquema nos permite realizar a atividade de compressão com um computador pessoal economizando até 35% de energia elétrica.

Este é um exemplo real de computação verde onde a mesma atividade é realizada por um mesmo equipamento, gastando muito menos energia e exposto apenas a uma pequena degradação de desempenho (em termos de *RD*).

Nosso esquema pode ser prontamente aplicado para constituir módulos de compressão de vídeo baseados em computadores pessoais, os quais serão capazes de ajustarem-se de forma adaptativa às condições de *RDE* sem a necessidade de se alterar a implementação do decodificador. Além disso, tecnologias como Cool'n Quiet[®] e SpeedStep[®] podem melhorar o desempenho do esquema proposto, permitindo com que o processador "durma" na medida em que espera pela chegada de quadros, isso quando a taxa de captura é menor que a velocidade de compressão proporcionada.

Algoritmos e implementações do mais novo padrão HEVC [19] ainda não estão maduros o suficiente para testes. Todavia, os conceitos aqui discutidos aplicam-se também ao HEVC.

Como nós provemos ajustes de um codificador de forma

a atingir alvos de taxa de bits e distorção, bem como a máxima velocidade de compressão, usando a mínima energia possível, esperamos estar contribuindo para o sistema mais "verde", reduzindo as emissões de carbono de servidores de computação intensiva.

REFERÊNCIAS

- [1] JVT of ISO/IEC MPEG and ITU-T VCEG, "Advanced video coding for generic audiovisual services," Tech. Rep. 14496-10:2005, March 2005.
- [2] I. E. G. Richardson, *H.264 and MPEG-4 Video Compression*, John Wiley & Sons Ltd, 2003.
- [3] R. L. de Queiroz, R. S. Ortis, A. Zaghetto, and T. A. Fonseca, "Fringe benefits of the H.264/AVC," in *Proc. International Telecommunication Symposium*, pp. 208–212, 2006.
- [4] S. Albers, "Energy-efficient algorithms," *Communications of the ACM*, vol. 53, no. 5, pp. 86–96, 2010.
- [5] K. Sayood, *Introduction to Data Compression*, Morgan Kaufmann Publishers, 2000.
- [6] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi, "Video coding with H.264/AVC: tools, performance, and complexity," *IEEE Circuits and Systems Magazine*, pp. 7–28, Jan-Mar 2004.
- [7] Y.-Y. Huang, B.-Y. Hsieh, S.-Y. Chien, S.-Y. Ma, and L.-G. Chen, "Analysis and complexity reduction of multiple reference frames motion estimation in H.264/AVC," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 16, no. 4, pp. 507–522, Apr. 2006.
- [8] Z. Chen, P. Zhou, and Y. He, "Hybrid unsymmetrical-cross multi-hexagon-grid search strategy for integer pel motion estimation in H.264," *Proc. Picture Coding Symposium*, Apr. 2003.
- [9] B.G. Kim, S.-K. Song, and C.-S. Cho, "Efficient inter-mode decision based on contextual prediction for the P-slice in H.264/AVC video coding," *Proc. IEEE Intl. Conf. on Image Processing*, pp. 1333–1336, September 2006.
- [10] C. S. Kannangara, Y. Zhao, I. E. Richardson, and M. Bystrom, "Complexity control of H.264 based on a Bayesian framework," *Proc. Picture Coding Symposium*, Nov. 2007.
- [11] Z. He, Y. Liang, L. Chen, I. Ahmad, and D. Wu, "Power-rate-distortion analysis for wireless video communication under energy constraints," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 15, no. 5, pp. 645–658, May 2005.
- [12] M. Moecke and R. Seara, "Sorting rates in video encoding process for complexity reduction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 1, pp. 88–101, 2010.
- [13] Kyu Park, Nitin Singhal, Man Hee Lee, and Sungdae Cho, "Efficient design and implementation of visual computing algorithms on the GPU," *Proc. IEEE Intl. Conf. on Image Processing*, pp. 2321–2324, Nov. 2009.
- [14] Intel, "Intel IPP," <http://software.intel.com/en-us/intel-ipp/>.
- [15] O. Silven and K. Jyrkkä, "Observations on power-efficiency trends in mobile communication devices," *EURASIP Journal on Embedded Systems*, vol. 2007, no. 1, pp. 17–27, 2007.
- [16] L. Merrit and R. Vanam, "Improved rate control and motion estimation for H.264 encoder," *Proc. IEEE Intl. Conf. on Image Processing*, vol. 5, 2007.
- [17] Gary J. Sullivan and Thomas Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 74–90, Nov. 1998.
- [18] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," *VCEG-M33*, April 2001.
- [19] G. Sullivan and J. Ohm, "Recent developments in standardization of high efficiency video coding (HEVC)," *Proc. SPIE Applications of Digital Image Processing XXXIII*, vol. 7798, 2010.