Estimação de Movimento com Medidas de Dispersão do Resíduo de Predição

Gabriel Lemes S. L. de Oliveira, Eduardo Peixoto e Ricardo L. de Queiroz

***Resumo*— A estimação de movimento por blocos é utilizada na maioria dos codecs de vídeos para explorar redundância temporal. Geralmente, avalia-se algum tipo de métrica de distância entre um bloco candidato e um bloco alvo, tais como a SAD ou a SSD. Neste artigo, argumentamos pela consideração de medidas de dispersão do resíduo junto com as métricas usuais. Como uma prova de conceito, um método é desenvolvido para levar medidas de dispersão em consideração no software de referência JM para o padrão H.264/AVC, mantendo absoluta compatibilidade com o padrão. Os resultados mostram melhorias significativas com relação ao codificador H.264 original.**

***Palavras-Chave*— Predição inter-frame, estimação de movimento, medidas de dispersão, codificação de vídeo, H.264/AVC.**

***Abstract*— Block motion estimation is used in most video codecs to exploit temporal redundancy between frames. It usually evaluates some sort of distance metrics such as the SAD or the SSD between a candidate block and a target block. In this paper, we argue for the consideration of dispersion measures of the residue along with the usual distortion metrics. As a proof of concept, a method is devised to bring dispersion measures into consideration in the H.264/AVC JM reference software while keeping it absolutely compliant to the standard. Results show significant improvements over the original H.264 encoder.**

***Keywords*— Inter-prediction, motion estimation, dispersion measures, video coding, H.264/AVC.**

I. Introdução

A maioria dos codecs de vídeo implementa alguma forma de predição inter-frame por estimação de movimento (ME, do inglês, *motion estimation*) baseada em blocos para explorar redundância temporal[1][2]. A fim de reduzir o número de bits necessários para codificar um bloco alvo em um frame a ser codificado, o codificador procura por um bloco de predição dentre um conjunto de blocos candidatos previamente codificados em frames anteriores. No lugar do bloco alvo, codifica-se então um vetor de movimento e um bloco de diferenças entre o bloco alvo e o bloco de predição, o resíduo, de forma que o decodificador seja capaz de reproduzir o bloco alvo. Esta técnica é implementada, por exemplo, no popular padrão H.264/AVC[3][4].

Em 2012, Blasi *et al* propuseram o método *predição inter-frame aprimorada* (EIP, do inglês, *enhanced inter-prediction*) para melhorar essa abordagem[5]. Este método consiste em transformar os blocos candidatos com uma transformação paramétrica inversível para melhorar sua correspondência ao bloco alvo. Para cada candidato, os parâmetros da transformação são otimizados nesse sentido. Os parâmetros ótimos para o candidato vencedor são então enviados juntos com o respectivo resíduo e demais informações de movimento para o decodificador, de maneira que ele seja capaz de inverter a transformação. A premissa subjacente ao EIP é que os bits extras necessários para codificar tais parâmetros são frequentemente compensados por resíduos menores, que então necessitariam de menos bits para serem codificados.

Como uma prova de conceito em favor do EIP, Blasi *et al* também propuseram a *transformação de deslocamento* (ST, do inglês, *shifting transformation*)[5]. A ST consiste em uma transformação de um único parâmetro que, por sua vez, consiste em uma única constante adicionada uniformemente a cada bloco. Para cada bloco candidato, a constante é otimizada no sentido de reduzir o resíduo resultante da comparação ao bloco alvo. A implementação do EIP com ST apresentada por Blasi *et al* no padrão H.264/AVC utilizando o software de referência JM[6] mostrou ganhos significativos com relação ao codificador H.264 sem modificações[5].

Neste artigo, propomos uma nova abordagem para a predição inter-frame, utilizando medidas de dispersão do resíduo para informar a estimação de movimento junto com as medidas usuais de distorção, tais como a soma das diferenças absolutas (SAD, do inglês, *sum of absolute differences*) ou a soma das diferenças quadradas (SSD, do inglês, *sum of squared differences*). Nossa proposta é fortemente influenciada pelo método EIP, particularmente pelo EIP com ST, e baseia-se na observação de que o EIP com ST é eficiente justamente porque tende a selecionar blocos candidatos que minimizam a dispersão do resíduo em algum sentido. Esta abordagem traz a significativa vantagem de não requerer a transmissão de informações de movimento extras, tais como os parâmetros ótimos de transformação da EIP. Isso significa que esta abordagem pode ser feita de maneira compatível com os padrões modernos de codificação de vídeo, i.e., sem ensejar mudanças no lado do decodificador. Como uma prova de conceito, desenvolvemos um técnica simples para ME informada por medidas de dispersão do resíduo e a integramos ao software de referência JM para o padrão H.264/AVC, mantendo absoluta compatibilidade com esse padrão.

II. EIP com Transformação de Deslocamento

Tipicamente, a estimação de movimento por blocos (ME) procura dentre frames previamente codificados por blocos de predição P que "melhor se ajustem"a um bloco alvo T em um frame a ser codificado. O codificador então fornece o vetor de movimento MV correspondente, de maneira que o decodificador consiga recriar a predição P. O resíduo $R = P-$

Gabriel Lemes S. L. de Oliveira, Eduardo Peixoto e Ricardo L. de Queiroz, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, Brasil, E-mails: gabriel.lemes@gmail.com, eduardopeixoto@ieee.org, queiroz@ieee.org. Este trabalho foi parcialmente financiado pelo CNPq e pela Capes.

T, ou uma aproximação dele caso seja permitida a codificação com perdas, é também codificado na bitstream de maneira que T, ou uma aproximação sua, possa ser recuperado. Um "melhor ajuste"para um dado alvo T é avaliado em termos de uma função de custo $cost(\cdot, T)$, geralmente dada pela SAD ou pela SSD, possivelmente pesando o custo de enviar o MV adequado. Mais precisamente, o codificador coloca na bitstream um vetor de movimento MV_o para um bloco de predição P_o, junto com o respectivo resíduo $R_o = P_o - T$, se P_o satisfaz $cost(P_o, T) \leq cost(P, T)$ para todo bloco candidato P considerado. O raciocínio por detrás desse esquema é que MV_o e R_o, geralmente, requerem menos bits para codificação a uma dada qualidade de imagem que o próprio T.

A predição inter-frame aprimorada (EIP) pretende aprimorar essa abordagem de ME ao considerar um conjunto de blocos candidatos *transformados* P', dados por

$$P' = \Theta(P|x^1, x^2, \dots x^n), \tag{1}$$

em que $\Theta(\cdot|x^1, x^2, \dots x^n)$ é uma transformação paramétrica *inversível* com parâmetros $\mathbf{x} = (x^1, x^2, \dots x^n)$. Para *cada* bloco candidato, $\mathbf{x}$ é fixado em $\mathbf{x}_o$ que minimiza o custo do candidato. Ou seja, para cada bloco candidato, $\mathbf{x}_o$ é tal que $cost(\Theta(P|\mathbf{x}_o), T) \leq cost(\Theta(P|\mathbf{x}), T)$ para todo vetor válido de parâmetros $\mathbf{x}$. Note que, se Θ torna-se a transformação identidade para algum $\mathbf{x}$ válido, então é sempre verdade que $cost(P', T) \leq cost(P, T)$, o que pode implicar em um resíduo $R' = P' - T$ menor e que requeira menos bits para codificação.

Uma vez que P'_o seja encontrado, o vetor $\mathbf{x}_o$ correspondente deve ser codificado junto com os respectivos R'_o e MV'_o. Assim, a técnica EIP so será efetiva se os bits necessários para codificar $\mathbf{x}_o$ forem compensados por resíduos que requeiram suficientemente menos bits para codificação. Além disso já que o vetor ótimo $\mathbf{x}_o$ é calculado para *todo* bloco candidato testado, a transformação $\Theta(\cdot|\mathbf{x})$ deve ainda ser tal que requeira pouco esforço computacional para o cálculo de $\mathbf{x}_o$, de maneira a manter praticável o custo computacional como um todo.

Assim, uma transformação particularmente efetiva para implementar a EIP é a transformação de deslocamento (ST)[5], que é uma transformação $\Theta(\cdot|s)$ simples, de parâmetro único s. O candidato transformado P' é dado simplesmente por

$$P' = \Theta(P|s) = P + s, \tag{2}$$

em que a soma é entendida no sentido de que s é uniformemente adicionado a cada elemento de P. A efetividade da EIP com ST deriva do fato de que existe um algoritmo simples para o cálculo de s_o e do fato de que s_o pode ser eficientemente codificado. Foi mostrado[5] que a EIP com ST pode ser integrada ao âmbito do H.264/AVC para melhorar significativamente o seu desempenho.

Procedemos agora a uma mostra da solução ótima pra s_o, uma vez que o nosso trabalho depende uma observação chave a respeito dessa solução. Para isso, no entanto, nos desviamos da demonstração original depois de segui-la de perto nos primeiros passos. Primeiramente, considere que o custo de um candidato P seja dado pela SAD com relação ao alvo T e que P, T, e $R = P - T$ consistem todos em blocos de $N = n \times m$ pixels, sendo n e m a largura e a altura dos blocos, respectivamente. O custo então é dado por

$$cost(P, T) = \sum_{i=1}^{N} |P(i) - T(i)| = \sum_{i=1}^{N} |R(i)|. \tag{3}$$

Uma vez que nem a SAD nem a forma de $\Theta(\cdot|s)$ dependem da ordem dos elementos de P, T ou R, assumimos também, sem perda de generalidade, que R seja ordenado de forma crescente, ou seja, $R(i) \leq R(j)$, $\forall i < j$. Considere agora, para P, T e R fixos, o custo de um bloco candidato deslocado por um parâmetro de deslocamento s:

$$cost(s) = \sum_{i=1}^{N} |\,(P(i) + s) - T(i)| = \sum_{i=1}^{N} |R(i) + s|. \tag{4}$$

Note que $\Theta(P|0) = P$, de forma que a transformação identidade é contemplada e $cost(P, T) = cost(0)$. Avaliamos então $cost(1)$, o custo de um bloco candidato com um deslocamento unitário *positivo*. Seja N_- o número de elementos negativos em R. De maneira análoga, sejam N_0 e N_+ os números de elementos nulos e de elementos positivos em R, respectivamente. Note que $N = N_- + N_0 + N_+$. O custo do candidato original pode então ser reescrito como

$$cost(0) = -\sum_{i=1}^{N_-} R(i) + \sum_{i=N_-+N_0+1}^{N} R(i), \tag{5}$$

enquanto $cost(1)$ é então dado por

$$cost(1) = -\sum_{i=1}^{N_-} (R(i) + 1) + \sum_{N_-+1}^{N_-+N_0} (1) + \sum_{i=N_-+N_0+1}^{N} (R(i) + 1)\,. \tag{6}$$

Depois de rearranjar a equação 6, finalmente temos

$$cost(1) = cost(0) - N_- + N_0 + N_+. \tag{7}$$

Comparando as equações 7 e 5, vemos que $cost(1) < cost(0)$ se, e somente se

$$N_- > N_0 + N_+. \tag{8}$$

É aqui que a nossa demonstração se afasta da original. Adicionando N_- a ambos os lados da inequação 8, vemos que um deslocamento unitário positivo reduzirá o custo do bloco candidato se, e somente se

$$N_- > \frac{N}{2}. \tag{9}$$

De maneira análoga, pode-se demonstrar que um deslocamento unitário *negativo* reduzirá o custo de um bloco candidato, i.e., $cost(-1) < cost(0)$, se, e somente se

$$N_+ > \frac{N}{2}. \tag{10}$$

Note que ambas as desigualdade 9 e 10 são desigualdades *estritas*.

Suponha agora que a desigualdade 9 seja verdade, o que significa que um deslocamento unitário positivo realmente reduzirá o custo do candidato ao mesmo tempo em que garante que a desigualdade 10 é falsa. Aplicamos então um deslocamento unitário positivo e ficamos com $P+1$ e $R+1$. Redefinimos N_-, N_0 a N_+ de maneira análoga à anterior, de acordo com o novo resíduo deslocado $R+1$. Suponha então que a condição 9 ainda seja satisfeita. Isso significa que a aplicação de um novo deslocamento unitário positivo reduzirá ainda mais o custo. Uma vez que

$$\Theta(\Theta(P, s_1), s_2) = \Theta(P, s_1 + s_2), \tag{11}$$

o que pode ser trivialmente demonstrado, isso também significa que

$$cost(2) < cost(1) < cost(0). \tag{12}$$

Podemos então iterar esse processo até que a condição 9 não seja mais satisfeita e ficamos então com o parâmetro ótimo de deslocamento s_o, necessariamente positivo, depois de s_o iterações. Fica garantido que, depois do último passo, a condição 10 *também* não será satisfeita. Caso contrário, a condição 9 não teria sido satisfeita antes do último passo e não teríamos chegado à s_o-ésima iteração. Poderíamos proceder de maneira similar para um deslocamento negativo caso a condição 10 tivesse sido verdadeira no princípio, o que tornaria a condição 9 falsa no princípio e resultaria em um deslocamento ótimo s_o negativo depois de $|s_o|$ iterações, quando então ambas as condições seriam falsas. Caso ambas as inequações sejam falsas desde o começo, $s_o = 0$.

Existem algoritmos mais eficientes que o descrito acima para o cálculo do parâmetro ótimo de deslocamento[5]. No entanto, ele nos provê uma informação chave. No valor ótimo do parâmetro de deslocamento, *ambas* as inequações 9 e 10 serão *falsas*. Supondo que N seja ímpar e lembrando que R está ordenado de forma crescente, isso significa que o elemento do meio em $R + s_o$ será *zero*. Ou seja, se N for ímpar, s_o é único e é dado pelo negativo da mediana de R. Se N for par, não existe mais unicidade na solução. Caso estejamos interessados no *menor* deslocamento $|s_o|$ capaz de produzir o custo ótimo, o que pode ser o caso em EIP já que esse deslocamento deverá ser codificado à parte, temos então que $s_o = -R(\frac{N}{2})$ se $R(\frac{N}{2}) > 0$, $s_o = -R(\frac{N}{2}+1)$ se $R(\frac{N}{2}+1) < 0$ e $s_o = 0$ caso contrário. No entanto, para N par, *qualquer* valor de s_o entre $-R(\frac{N}{2})$ e $-R(\frac{N}{2}+1)$ resultará no exato mesmo custo. O negativo da mediana de R certamente encontra-se nesse intervalo. Assim, independente da paridade de N, podemos dizer que

$$s_o = -\tilde{R}, \tag{13}$$

em que $\tilde{R}$ é a mediana de R, é um valor ótimo para o parâmetro de deslocamento s_o.

Considere agora que o custo seja dado pela SSD. Analogamente, o custo para um candidato deslocado será dado então por

$$cost(s) = \sum_{i=1}^{N}((P(i)+s) - T(i))^2 = \sum_{i=1}^{N}(R(i)+s)^2. \tag{14}$$

Derivando a equação 14 com respeito a s e igualando o resultado aplicado em $s = s_o$ a zero, temos

$$s_o = -\frac{\sum_{i=1}^{N} R(i)}{N} = -\bar{R}, \tag{15}$$

de forma que o deslocamento ótimo é simplesmente o negativo da média dos valores do resíduo R.

III. Considerações Heurísticas para ME com Medidas de Dispersão

A equação 15 para o parâmetro ótimo de deslocamento no caso da SSD como medida de custo já havia sido dada em uma forma levemente diferente no trabalho original de Blasi *et al*. No entanto, a equação 13 para o parâmetro ótimo de deslocamento no caso da SAD como medida de custo é um tanto mais difícil de perceber a partir do trabalho original e, por isso, resolvemos demonstrá-la diretamente na seção anterior. Acreditamos que ela revela bastante sobre o *porque* do método EIP com ST ser eficiente.

O valor mediano de uma amostra é uma medida da tendência central daquela amostra. Olhando para a SAD do bloco candidato deslocado,

$$SAD_{shift} = \sum_{i=1}^{N} |R(i) - \tilde{R}|, \tag{16}$$

percebemos que ela é proporcional ao desvio médio com relação àquela medida de tendência central. O valor médio de uma amostra é, também, uma medida da sua tendência central e, mais uma vez, olhando para a SSD deslocada, vemos que

$$SSD_{shift} = \sum_{i=1}^{N} \left(R(i) - \bar{R}\right)^2, \tag{17}$$

que é proporcional à variância, também uma medida do desvio médio da amostra com relação à sua tendência central. As equações 16 e 17 são ambas medidas de *dispersão* do resíduo do bloco candidato. De qualquer forma, note que o método EIP com ST termina por dar preferência para candidatos que minimizem a dispersão do resíduo.

São duas a principais vantagens que enxergamos em escolher uma predição que resulte em resíduos com baixa dispersão. Primeiro, em diversos codecs modernos, o bloco de resíduo é primeiramente transformado por alguma transformada do tipo DCT e só depois quantizado e codificado na bitstream[1][2][4]. Quando utilizamos a equação 16 ou a equação 17 para escolher um bloco de predição, o coeficiente DC perde sua importância relativa nessa escolha, de forma que os coeficientes AC do bloco alvo são melhor aproximados antes da quantização. Isso pode implicar em menos coeficientes não nulos a serem codificados e, possivelmente, uma perda menor de detalhes de textura. Segundo, uma dispersão menor implica uma entropia menor, o que também pode implicar um número menor de bits necessários para a

codificação do bloco. Essas são nossas principais razões para acreditar que a dispersão do resíduo fornece uma poderosa peça de informação que deve ser considerada na estimação de movimento.

IV. Considerações Práticas e Implementação no Padrão H.264/AVC

Para mostrar que a consideração de medidas de dispersão em ME pode, de fato, aprimorar a eficiência de codificação em compressão de vídeos, desenvolvemos uma técnica simples a ser integrada no software de referência JM[6] do padrão H.264/AVC. A técnica proposta é estritamente compatível com o padrão, de forma que nenhuma adaptação é necessária no lado do decodificador.

Como a medida de dispersão a ser utilizada, propomos o *desvio absoluto total com relação à média* (DATM), dado por

$$DATM = \sum_{i=1}^{N} |R(i) - \bar{R}|. \tag{18}$$

Escolhemos essa medida por sua simplicidade. Note que ela não requer nem a operação de ordenamento da equação 16, necessária para encontrar a mediana, nem a elevação de cada termo ao quadrado da equação 17.

Nossos testes iniciais com a simples substituição da SAD pela DATM mostraram que essa abordagem ingênua falha, principalmente, por causa de decisões ineficientes sobre os modos de macrobaloco. A técnica que propomos, então, consiste em uma solução simples para essa abordagem ingênua. Duas predições são feitas para cada cada macrobloco. Uma utilizando as funções de distorção originais do codificador H.264 e outra com a DATM no lugar delas. O codificador então coloca na stream a melhor delas, no sentido taxa-distorção. Nenhuma outra funcionalidade do codificador H.264 é alterada, mesmo durante a passada com DATM. Na nossa implementação, o tempo de codificação é essencialmente dobrado, mas note que a técnica é naturalmente paralelizável em sua essência.

O tamanho de um macrobloco no padrão H.264/AVC é fixado em 16×16 pixels. Para cada macrobloco, a ME é feita para subpartições de tamanho 16×16, 16×8, 8×16, 8×8, 8×4, 4×8 ou 4×4. Para a codificação do resíduo, no entanto, cada macrobloco é dividido em partições de tamanho 4×4 que são então transformadas por uma aproximação inteira da DCT e então quantizadas, independentemente dos tamanhos das subpartições efetivamente utilizadas durante a ME. Com isso em mente, então, a nossa implementação, na verdade, mede a dispersão de cada sub-bloco de tamanho 4×4 dentro de cada bloco de predição como em

$$DATM = \sum_{j=1}^{N_{sb}} \sum_{i=1}^{16} |R_j(i) - \bar{R_j}|, \tag{19}$$

em que $N_{sb} = \frac{N}{16}$ é o número de sub-blocos de tamanho 4×4 em cada bloco de predição candidato e $\bar{R}_j$ é a média do j-ésimo sub-bloco de tamanho 4×4, R_j, dentro de cada bloco de predição candidato. O padrão H.264/AVC também permite a utilização opcional de uma transformada DCT inteira em partições de tamanho 8×8 para a quantização e codificação do resíduo. Nesse caso, a DATM da equação 19 é analogamente adaptada para sub-blocos de tamanho 8×8.

V. Resultados

Nossa implementação foi testada em algumas sequências populares utilizando toda a sua extensão. O codificador foi configurado para utilizar a estrutura de codificação IPPP com cinco frames de referência e com o codificador de entropia CABAC. Cada sequência foi testada em quatro QP's diferentes, especificamente, 22, 27, 33 e 37. Os resultados são mostrados na Tabela I em termos da BD-rate[7]. O codificador H.264 original com configurações idênticas foi utilizado como âncora para todos os testes.

A técnica proposta supera o codificador JM em todas as sequências testadas. Os resultados mostram ganhos de até 3,70% e de pelo menos 1,32% nessas sequências, com um ganho médio de 2,5%. Os resultados mostram também que a consideração da DATM leva a ganhos consistentes em todas as faixas de taxas, embora ganhos maiores sejam obtidos na faixa de altas taxas para a maioria das sequências testadas

TABELA I
BD-RATE DO DATM CONTRA O H.264/AVC CONVENCIONAL.

Resolução	FPS	Sequência	%bit	%altas	%baixas
704×480	60	Crew	-3,25	-3,12	-2,80
832×480	50	PartyScene	-1,32	-1,10	-1,51
1920×1080	50	Cactus	-3,07	-3,43	-2,43
1920×1080	50	BasketballDrive	-2,17	-2,39	-1,91
1920×1080	24	Kimono1	-3,70	-4,31	-3,05
1920×1080	24	ParkScene	-1,46	-1,56	-1,40

VI. Conclusões

Mostramos os benefícios da ME informada pela dispersão dos valores do resíduo. Notamos que o método EIP com ST já fornece um forte testemunho sobre esses benefícios. Para consolidar a importância das medidas de dispersão durante a ME, apresentamos ainda uma técnica em duas passadas para integrar uma medida de dispersão à operação de ME do padrão H.264/AVC de codificação de vídeo. A técnica proposta afeta somente as decisões do codificador a respeito dos vetores de movimento e dos modos de macrobloco, de forma que a bitstream gerada é absolutamente compatível com a formatação ditada pelo padrão H.264/AVC. Isso implica que nenhuma alteração do lado do decodificador é requerida para a sua implementação adequada. A DATM foi proposta como medida de distorção pela sua simplicidade. Os resultados mostram melhorias significativas sobre o codificador JM original com um ganho médio de 2,5% em termos de BD-rate, fornecendo um apoio adicional à nossa afirmação de que a ME pode ser aprimorada com a consideração da dispersão do resíduo. É possível que utilizações mais sofisticadas da DATM ou de outras medidas de dispersão possam prover ganhos ainda maiores, mas a técnica proposta em particular talvez seja interessante por ela mesma em diversas aplicações, dadas a sua simplicidade e a sua compatibilidade com o popular padrão H.264/AVC.

Em trabalhos futuros, pretendemos investigar outras medidas de dispersão bem como outras maneiras de integrá-las à operação de ME. Pretendemos ainda avaliar a integração dessas técnicas ao padrão emergente HEVC[8].

REFERÊNCIAS

[1] K. Sayood, *Introduction to Data Compression*. Morgan Kaufmann, 2012.

[2] D. Salomon, *Data Compression, The Complete Reference*. Springer, 2006.

[3] ITU-T, *ITU-T recommendation H.264, Advanced Video Coding for Generic Audiovisual Services*. ITU-T, 2014.

[4] I. E. Richardson, *The H.264 Advanced Video Compression Standard*. Wiley, 2010.

[5] S. G. Blasi, E. Peixoto and E. Izquierdo, "Enhanced Inter-Prediction via Shifting Transformation in the H.264/AVC," *IEEE Transactions on Circuits and Systems for Video Technology*, v. 23, n. 4, pp. 735–740, 2012.

[6] Joint Model (JM), "H.264/AVC Reference Software," *Artech House Inc.*, http://iphome.hhi.de/suehring/tml/download

[7] Gisle Bjontegaard, "Improvements of the BD-PSNR model," ITU-T SG16/Q6, 35th VCEG Meeting, Doc.VCEG-AI11, 2008.

[8] ITU-T, *ITU-T recommendation H.265, High Efficiency Video Coding*. ITU-T, 2014.