



**Juliano Melquiades Vianello**

**Métodos de Estimação de Movimento com Resolução em  
Subpixel no CODEC H264/AVC**

**Dissertação de Mestrado**

Dissertação apresentada como requisito parcial para  
obtenção do título de Mestre pelo Programa de Pós-  
Graduação em Engenharia Elétrica do Centro  
Técnico Científico da PUC-Rio.

Orientador: Prof. Marco Antonio Grivet Mattoso Maia

Rio de Janeiro  
Fevereiro de 2007



**Juliano Melquiades Vianello**

**Métodos de Estimação de Movimento com  
Resolução em Subpixel no Codec H264/AVC**

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-Graduação em Engenharia Elétrica do Departamento de Engenharia Elétrica do Centro Técnico Científico da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

**Prof. Marco Antonio Grivet Mattoso Maia**

Orientador

Centro de Estudos em Telecomunicações - PUC-Rio

**Prof. Carla Liberal Pagliari**

IME

**Prof. Marcelo de Mello Perez**

IME

**Prof. Weiler Alves Finamore**

Centro de Estudos em Telecomunicações - PUC-Rio

**Prof. José Eugenio Leal**

Coordenador Setorial do Centro Técnico Científico - PUC-Rio

Rio de Janeiro, 14 de fevereiro de 2007

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

### **Juliano Melquiades Vianello**

Graduou-se em Engenharia de Telecomunicações no IME (Instituto Militar de Engenharia) em 2003. Atuou no projeto de automação de laboratórios do Centro Brasileiro de Pesquisas Físicas (CBPF), na gerência da Rede Rio de Computadores e na área de gerência de projetos da EMBRATEL. Atualmente é Chefe da Seção de Comunicações e Eletrônica do Centro de Avaliações do Exército.

#### Ficha Catalográfica

Vianello, Juliano Melquiades

Métodos de estimação de movimento com resolução em subpixel no CODEC H264/AVC : implementação e comparação com outros métodos / Juliano Melquiades Vianello ; orientador: Marco Antonio Grivet Mattoso Maia. – 2006.

146 f. : il. ; 30 cm

Dissertação (Mestrado em Engenharia Elétrica)– Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2007.

Inclui bibliografia

1. Engenharia elétrica – Teses. 2. Codificação de vídeo. 3. Estimação de movimento. 4. Algoritmo rápido. 5. CODEC H264/AVC. 6. Pesquisa de subpixel. 7. Redução do tempo de estimação de movimento. I. Maia, Marco Antonio Grivet Mattoso. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Elétrica. III. Título.

CDD: 623.1

Para Deus, minha família e Priscilla,  
pelo apoio.

## Agradecimentos

À Deus, por tudo.

Ao meu orientador Professor Marco Antonio Grivet Mattoso Maia pelo estímulo e parceria para a realização deste trabalho.

À PUC-Rio, pelos auxílios concedidos, sem os quais este trabalho não poderia ter sido realizado.

Aos meus pais, pela educação, atenção e carinho de todas as horas.

À minha namorada, Priscilla, por todo o apoio e paciência.

À minha segunda família, Zeca, Vânia e Renato, pelas palavras de incentivo.

Ao meus amigos Dalto, Artur e Leonardo por toda ajuda.

Aos meus colegas da PUC-Rio.

Aos professores que participaram da Comissão examinadora, em especial a Professora Carla e Professor Perez pelas palavras de incentivo.

A todos os professores e funcionários do Departamento pelos ensinamentos e pela ajuda.

A todos os amigos e familiares que de uma forma ou de outra me estimularam ou me ajudaram.

## Resumo

Vianello, Juliano Melquiades. **Método de estimação de movimento com resolução em subpixel no CODEC H264/AVC**. Rio de Janeiro, 2007. 146p. Dissertação de Mestrado - Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

Foi desenvolvido pelos grupos MPEG (Moving Picture Expert Group) e VCEG (Video Coding Expert Group) um novo padrão de codificação de vídeo denominado AVC (Advanced Vídeo Coding). Este padrão fornece uma capacidade de compressão maior de vídeo se comparado com os padrões anteriores. A estimação de movimento é a fase da codificação de vídeo que demanda maior tempo de processamento. Estimação de movimento é computacionalmente custosa para H.264/AVC se o método Full Search [1] é usado. A fim de reduzir o tempo de codificação, o software de referência JM 9.8 adota um rápido método de estimação de movimento para pixel inteiro chamado UMHexagonS e um método para pixel fracionário chamado CBFPS. Um método proposto por Xiaoquan Yi, Jun Zhang, Nam Ling e Weijia Shang [2] chamado aqui, por simplicidade, de P021 apresenta, em comparação com o software de referência JM, uma melhor e simplificada forma de estimação de movimento para aumentar a velocidade do processo de codificação e manter a qualidade visual do vídeo. Visando diminuir ainda mais o tempo de processamento e o custo computacional apresentada pelos métodos citados anteriormente, o que é necessário principalmente para aplicações de codificação de vídeo móveis, como celulares e palm tops e aplicações de tempo real, como videoconferência, este trabalho propõe um método de estimação de movimento que consiste em evitar o processamento de blocos que não se beneficiam da pesquisa de metade de pixel. Após a execução dos testes, pode-se constatar que com uma qualidade de vídeo semelhante, o método proposto reduz o tempo de processamento em 76,17%, 75,95% e 11,74% em média quando comparado com os métodos Full Search, JM 98 e P021. Além disso, este método praticamente preserva a taxa de bits, apresentando um pequeno aumento de 8,72% , 8,06% e 8,02% em média quando comparado com os mesmos métodos. Isto representa, em nossa opinião, um pequeno preço a ser pago, considerando os benefícios em termos de tempo de processamento.

## **Palavras-chave**

codificação de vídeo, estimação de movimento, algoritmo rápido, CODEC H264/AVC, pesquisa de subpixel, redução do tempo de estimação de movimento.

## Abstract

Vianello, Juliano Melquiades. Motion Estimation Method with subpixel resolution of Codec H264/AVC. Rio de Janeiro, 2007. 146p. MSc Dissertation - Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

The MPEG (Moving Picture Experts Group) and the VCEG (Video Coding Experts Group) have developed a new video compression standard entitled AVC (Advanced Video Coding). This standard offers the capacity of video compression greater than the previous standards. The motion estimation which is a phase of the video compression is extremely computer-intensive and therefore demands most of the processing time. When Full Search [1] method is used for H.264/AVC, this process is extremely expensive. In order to reduce encoding time, the reference software JM (throughout this work, JM98 is the version used) has respectively adopted a fast motion estimation method for the integer pixel called UMHexagonS and a method for the sub-pixel called CBFPS. A method proposed by Xiaoquan Yi, Jun Zhang, Nam Ling e Weijia Shang [2] called here P021 (as referenced by the Joint Video Team) has shown, when compared to the reference software JM, a simple way of motion estimation that increases the speed of coding process while maintaining the video visual quality. In order to decrease the processing time and computational cost of these methods, which are: i) particularly needed in mobile video coding applications such as mobile phones and palmtops; ii) in real time applications such as videoconference, this work proposes a motion estimation method that eliminates the processing of blocks that does not produce benefits for the half-pixel search. The simulation results show that the proposed method reduces the processing time in 76,17%, 75,95% and 11,74% in average when respectively compared with Full Search, JM98 and P021 methods, without relevant impact in video quality. Besides, this method produces a bits rate increase of 8,72% , 8,06% e 8,02% in average when compared with the same methods. This is in our opinion, an inexpensive price to be paid when the time benefits are considered.



## **Keywords**

Video coding, motion estimation, fast algorithm, Codec H264/AVC, subpixel search, reduced of the motion estimation time.

## Sumário

|  |    |
|--|----|
| 1 Introdução   | 17 |
| 2 Conceitos iniciais   | 19 |
| 2.1. Tipos de formatos de seqüências de vídeo                  | 19 |
| 2.2. Medida da Qualidade de vídeo                              | 21 |
| 2.2.1 Qualidade Subjetiva                                      | 22 |
| 2.2.2 Qualidade Objetiva                                       | 23 |
| 2.3 Codificação de vídeo                                       | 24 |
| 2.4 Codificadores e Decodificadores de vídeo                   | 26 |
| 2.5 Energia de um bloco residual                               | 29 |
| 2.6 Conceito de Macrobloco                                     | 33 |
| 2.7 Resumo e conclusão do capítulo                             | 34 |
| 3 Estimação e Compensação de movimento na codificação de vídeo | 36 |
| 3.1 Resíduo formado pela diferença entre quadros               | 36 |
| 3.2 Fluxo óptico   | 37 |
| 3.3 Estimação e Compensação de movimento baseada em bloco      | 40 |
| 3.4 Tamanho de bloco variável                                  | 42 |
| 3.5 Compensação de movimento com precisão de subpixel          | 46 |
| 3.6 Resumo e conclusão do capítulo                             | 49 |
| 4 H.264  | 51 |
| 4.1 Conceitos importantes                                      | 51 |
| 4.2 Codificadores e Decodificadores H.264:                     | 53 |
| 4.2.1 Codificador H.264  | 53 |
| 4.2.2 Decodificador H.264                                      | 54 |
| 4.3 Perfis do H.264  | 55 |
| 4.4 Modos de inter predição no H.264                           | 56 |
| 4.5 Medidas de desempenho                                      | 56 |

|   |     |
|---|-----|
| 4.5.1 Desempenho quanto a taxa-distorção  | 57  |
| 4.5.2 Desempenho computacional  | 60  |
| 4.6 Resumo e conclusão do capítulo  | 63  |
| 5 Métodos rápidos utilizados para estimação de movimento  | 65  |
| 5.1 Full Search   | 65  |
| 5.2 O software de referência JM 9.8   | 67  |
| 5.2.1 Algoritmo UMHexagonS para estimação de movimento de pixel inteiro                               | 68  |
| 5.2.2 Algoritmo CBFPS para estimação de movimento de pixel fracionário                                | 69  |
| 5.3 P021 - Método de Xiaoquan Yi, Jun Zhang, Nam Ling e Weijia Shang                                  | 70  |
| 5.3.1 Descrição do método P021  | 71  |
| 5.3.1.1 Utilização de pouca memória   | 71  |
| 5.3.1.2 Pesquisa rápida de estimação de movimento de pixel inteiro                                    | 72  |
| 5.3.1.3 Pesquisa rápida de estimação de movimento fracionária   | 74  |
| 5.4 Método proposto   | 78  |
| 5.4.1 Taxa de pesquisa efetiva (ESR) e Limite ótimo de SAD  | 80  |
| 5.4.2 Relação entre ESR e OSR para seqüências de vídeo e a primeira modificação do algoritmo proposto | 81  |
| 5.4.3 Ajuste dinâmico do limite de SAD e a segunda modificação do algoritmo proposto                  | 83  |
| 5.5 Resumo e conclusão do capítulo  | 87  |
| 6 Resultados da Simulação   | 89  |
| 6.1 Parâmetros da simulação   | 89  |
| 6.2 Sequências de vídeo   | 91  |
| 6.3 Desempenho computacional  | 95  |
| 6.4 Desempenho em termos da taxa-distorção  | 96  |
| 6.5 Resumo e conclusão do capítulo  | 102 |
| 7 Conclusão   | 103 |
| 8 Referências Bibliográficas  | 105 |

## Lista de figuras

|  |    |
|--|----|
| Figura 1- Quadros de vídeo amostrados nas resoluções: 4CIF, CIF, QCIF e SQCIF.     | 21 |
| Figura 2 - Exemplos de PSNR: (a) original; (b) 30.6 dB; (c) 28.3 dB                | 23 |
| Figura 3-Imagem com fundo borrado (PSNR = 27.7 dB)                                 | 24 |
| Figura 4- Correlação Espacial e temporal numa seqüência de vídeo                   | 26 |
| Figura 5- Diagrama de blocos de um vídeo codificado.                               | 27 |
| Figura 6-(a) Bloco 8x8 da matriz de luminância Y<br>(b)coeficientes transformados. | 28 |
| Figura 7 – Cálculo dos coeficientes transformados e quantizados                    | 28 |
| Figura 8- Região de pesquisa no frame anterior (referência)                        | 31 |
| Figura 9- Mapa MSE   | 31 |
| Figura 10- Mapa MAE  | 32 |
| Figura 11- Mapa SAE (SAD)  | 32 |
| Figura 12- Macrobloco (4:2:0)  | 34 |
| Figura 13- Quadro 1: preditor  | 38 |
| Figura 14- Quadro 2: quadro atual  | 38 |
| Figura 15- Diferença entre os quadros 1 e 2: quadro residual                       | 39 |
| Figura 16- Fluxo óptico  | 39 |
| Figura 17- Estimação de movimento  | 40 |
| Figura 18- Quadro 1  | 43 |
| Figura 19- Quadro 2  | 44 |
| Figura 20- Residuo (sem compensação de movimento)                                  | 44 |
| Figura 21- Residuo (com compensação de movimento e tamanho de bloco 16x16)         | 45 |
| Figura 22- Residuo (com compensação de movimento e tamanho de bloco 8x8)           | 45 |

|  |    |
|--|----|
| Figura 23- Resíduo (com compensação de movimento e tamanho de bloco 4x4)                                       | 46 |
| Figura 24- Estimação de movimento de pixel inteiro, metade e um quarto   | 47 |
| Figura 25- Bloco residual (bloco 4x4, compensação com precisão de meio-pixel)                                  | 48 |
| Figura 26- Bloco residual (bloco 4x4, compensação com precisão de um quarto de pixel)                          | 48 |
| Figura 27- Codificador H264  | 54 |
| Figura 28- Decodificador H.264   | 55 |
| Figura 29- Diferentes tamanhos de partição em um macrobloco e modos de predição no H.264                       | 56 |
| Figura 30- Comparação do desempenho quanto à taxa-distorção para a seqüência 'Office', CIF                     | 58 |
| Figura 31- Comparação do desempenho quanto à taxa-distorção para a seqüência 'Grasses', CIF                    | 58 |
| Figura 32- Frame original da seqüência 'Office'  | 59 |
| Figura 33- Frame original da seqüência 'Grasses'   | 59 |
| Figura 34- Imagem mostrando ruído em uma câmera (metade inferior)  | 62 |
| Figura 35- Método Full search com ordem de pesquisa em "varredura"   | 66 |
| Figura 36- Método Full Search com ordem de pesquisa em espiral   | 67 |
| Figura 37- Processo de pesquisa do algoritmo UMHExagonS  | 69 |
| Figura 38 – Implementação do algoritmo CBFPS   | 70 |
| Figura 39 - Fluxograma da pesquisa rápida de pixel inteiro do método P021                                      | 73 |
| Figura 40- Fluxograma do algoritmo de referência de pesquisa de subpixel.                                      | 74 |
| Figura 41- Fluxograma do algoritmo P021 de pesquisa de subpixel.   | 75 |
| Figura 42- Fluxograma da pesquisa de subpixel rápida completa (fast full sub-pel search) usada no método P021. | 76 |
| Figura 43 - Fluxograma da pesquisa rápida de subpixel usada no método P021. MVmin: MV tem o custo mínimo.      | 77 |
| Figura 44-Diminuição do PSNR para um limite de SAD maior.  | 81 |

|   |     |
|---|-----|
| Figura 45- Um modelo linear de ESR e OSR  | 82  |
| Figura 46 – Média dos tempos totais de processamento para cada fator multiplicador da Eq. 5.5.  | 87  |
| Figura 47 - Quadro de cada uma das seqüências de vídeo utilizadas nos testes: (a) akiyo (b) foreman (c) container (d) grandma (e) mother_daughter (f) suzie (g) coastguard (h) hall_monitor (i) husky (j) mobile. | 94  |
| Figura 48- Desempenho em termos de taxa-distorção para a seqüência de vídeo akiyo, com 100 quadros, CAVLC, freqüência de 10 Hz, IPPP, RDO=0.  | 96  |
| Figura 49- Desempenho em termos de taxa-distorção para a seqüência de vídeo akiyo, com 100 quadros, CABAC, freqüência de 30 Hz, IBBP, RDO=0.  | 97  |
| Figura 50- Desempenho em termos de taxa-distorção para a seqüência de vídeo akiyo, com 100 quadros, CAVLC, freqüência de 10 Hz, IPPP, RDO=1.  | 99  |
| Figura 51- Desempenho em termos de taxa-distorção para a seqüência de vídeo akiyo, com 100 quadros, CAVLC, freqüência de 10 Hz, IPPP, RDO=2.  | 100 |

## Lista de tabelas

|   |     |
|---|-----|
| Tabela 1- Formatos de vídeo   | 20  |
| Tabela 2- Parâmetros ITU-R BT.601-5   | 20  |
| Tabela 3- Desempenho computacional do H.264 (Violin, QCF, 25 frames)                                  | 61  |
| Tabela 4- ESR e OSR   | 82  |
| Tabela 5 – Valores de ESR, ASR e OSR para os 30 primeiros quadros codificados da seqüência akiyo.qcif | 86  |
| Tabela 6 - Seqüências de vídeo utilizadas na execução dos testes e suas propriedades.                 | 92  |
| Tabela 7- Sumário dos resultados das comparações  | 108 |
| Tabela 8.1 - Resultado da simulação do método Full Search para RDO=0 e IPPP                           | 109 |
| Tabela 8.2- Resultado da simulação do método Proposto para RDO=0 e IPPP                               | 110 |
| Tabela 8.3- Resultado da comparação entre o método Full Search e Proposto para RDO=0 e IPPP           | 111 |
| Tabela 9.1- Resultado da simulação do método JM9.8 para RDO=0 e IPPP                                  | 112 |
| Tabela 9.2- Resultado da simulação do método Proposto para RDO=0 e IPPP                               | 113 |
| Tabela 9.3- Resultado da comparação entre o método JM9.8 e Proposto para RDO=0 e IPPP                 | 114 |
| Tabela 10.1- Resultado da simulação do método P021 para RDO=0 e IPPP                                  | 115 |
| Tabela 10.2- Resultado da simulação do método Proposto para RDO=0 e IPPP                              | 116 |
| Tabela 10.3- Resultado da comparação entre o método P021 e Proposto para RDO=0 e IPPP                 | 117 |
| Tabela 11.1- Resultado da simulação do método Full Search para RDO=0 e IBBP                           | 118 |
| Tabela 11.2- Resultado da simulação do método Proposto para RDO=0 e IBBP                              | 119 |
| Tabela 11.3- Resultado da comparação entre o método Full Search e Proposto para RDO=0 e IBBP          | 120 |
| Tabela 12.1- Resultado da simulação do método JM9.8 para RDO=0 e IBBP                                 | 121 |
| Tabela 12.2- Resultado da simulação do método Proposto para RDO=0 e IBBP                              | 122 |
| Tabela 12.3- Resultado da comparação entre o método JM9.8 e Proposto para RDO=0 e IBBP                | 123 |
| Tabela 13.1- Resultado da simulação do método P021 para RDO=0 e IPPP                                  | 124 |
| Tabela 13.2- Resultado da simulação do método Proposto para RDO=0 e IPPP                              | 125 |

|  |     |
|--|-----|
| Tabela 13.3- Resultado da comparação entre o método P021 e Proposto para RDO=0 e IBBP        | 126 |
| Tabela 14.1- Resultado da simulação do método Full Search para RDO=1 e IPPP                  | 127 |
| Tabela 14.2- Resultado da simulação do método Proposto para RDO=1 e IPPP                     | 128 |
| Tabela 14.3- Resultado da comparação entre o método Full Search e Proposto para RDO=1 e IPPP | 129 |
| Tabela 15.1- Resultado da simulação do método JM9.8 para RDO=1 e IPPP                        | 130 |
| Tabela 15.2- Resultado da simulação do método Proposto para RDO=1 e IPPP                     | 131 |
| Tabela 15.3- Resultado da comparação entre o método JM9.8 e Proposto para RDO=1 e IPPP       | 132 |
| Tabela 16.1- Resultado da simulação do método P021 para RDO=1 e IPPP                         | 133 |
| Tabela 16.2- Resultado da simulação do método Proposto para RDO=1 e IPPP                     | 134 |
| Tabela 16.3- Resultado da comparação entre o método P021 e Proposto para RDO=1 e IPPP        | 135 |
| Tabela 17.1- Resultado da simulação do método Full Search para RDO=2 e IPPP                  | 136 |
| Tabela 17.2- Resultado da simulação do método Proposto para RDO=2 e IPPP                     | 137 |
| Tabela 17.3- Resultado da comparação entre o método Full Search e Proposto para RDO=2 e IPPP | 138 |
| Tabela 18.1- Resultado da simulação do método JM9.8 para RDO=2 e IPPP                        | 139 |
| Tabela 18.2- Resultado da simulação do método Proposto para RDO=2 e IPPP                     | 140 |
| Tabela 18.3- Resultado da comparação entre o método JM9.8 e Proposto para RDO=2 e IPPP       | 141 |
| Tabela 19.1- Resultado da simulação do método P021 para RDO=2 e IPPP                         | 142 |
| Tabela 19.2- Resultado da simulação do método Proposto para RDO=2 e IPPP                     | 143 |
| Tabela 19.3- Resultado da comparação entre o método P021 e Proposto para RDO=2 e IPPP        | 144 |
| Tabela 20- Parâmetros utilizados nas Tabelas 9 a 21 com sua descrição e unidade de medida.   | 145 |