

## 6

## Análise dos Resultados

### 6.1

#### Introdução

Este capítulo apresenta os resultados alcançados através da avaliação comparativa entre os algoritmos FSA (*Full Search Algorithm*), UMHexagonS (*Unsymmetrical-cross Multi-Hexagon-grid Search*) e o algoritmo proposto neste trabalho AUMHexagonS (*Adaptive Unsymmetrical-cross Multi-Hexagon-grid Search*).

Primeiramente detalhamos o ambiente e os parâmetros utilizados para as simulações para posteriormente apresentarmos os resultados obtidos com o algoritmo proposto.

### 6.2

#### Ambiente das Simulações

Todas as simulações foram realizadas no seguinte ambiente de desenvolvimento:

- Codificador do padrão H.264/AVC: versão JM11.0 fornecido pelo JVT;
- Computador: LapTop HP Compaq nx6320;
- Sistema Operacional: Microsoft Windows XP Professional (5.1, Build 2600);
- Processador: Intel CPU T2300 @ 1.66GHz (2 CPUs);
- Memória: 980MHz e 1016MB;
- Características do Software Utilizado: Compilador C para Windows (MS Visual Studio).

## 6.3

### Parâmetros das Simulações

Foram definidos dois conjuntos de parâmetros, para as simulações realizadas, os quais foram escolhidos levando-se em consideração:

- Algumas recomendações do órgão JVT, através do documento VCEG-N81 [29], com o objetivo de facilitar análises comparativas entre os algoritmos submetidos para o JVT;
- Utilizar parâmetros que evidenciem as modificações introduzidas;
- Utilizar parâmetros comuns nas simulações referentes à compressão de vídeo.

Seguindo as premissas levantadas através dos itens citados anteriormente, os conjuntos de parâmetros utilizados nas simulações deste trabalho foram:

#### 6.3.1

##### Primeiro Conjunto de Parâmetros

Esse primeiro conjunto de parâmetros foi utilizado nas simulações com seqüências QCIF e HDTV e levou em consideração parâmetros recomendados pelo JVT, parâmetros utilizados com o objetivo de evidenciar as modificações introduzidas e parâmetros comuns nas simulações referentes à compressão de vídeo.

Foram realizadas 48 simulações com este conjunto de parâmetros, descritos a seguir:

1) Parâmetros recomendados pelo JVT:

- Resolução de vídeo: QCIF (176x144) e HDTV (1280x720);
- Codificação por Entropia: CABAC;
- Números de quadros usados como referência: 1;
- Hadamard: ligada
- Janela de Busca (W): 32
- *RDOptimization mode decision*: 1 (ligado no modo *High complexity*);
- Passos de Quantização (QP): 16, 20, 24 e 28 (incluindo o 1º quadro);
- Taxa de quadros: 30Hz para QCIF e 60Hz para HDTV (constantes durante toda a seqüência);

- Estrutura do GOP: IBBP;
- Espaço de Cor: 4:2:0.

2) Parâmetros utilizados com o objetivo de evidenciar as modificações introduzidas:

- Sequências de vídeo: Utilizar sequências que possuam cenas com alto, médio e baixo movimento. Dessa maneira foram escolhidas as sequências QCIF *Walk*, *Foreman* e *Container* e HDTV *Stockholm*. Para a sequência HDTV foram selecionados quadros com movimentos médios e baixos;
- Algoritmos: Utilizar os algoritmos implementados no código do H.264/AVC. Dessa forma foram escolhidos os algoritmos FSA e UMHS para as análises em conjunto com o AUMHS;
- Acuracidade do Pixel: Pixel inteiro, pois as modificações introduzidas estão na parte do código de estimação de movimento para pixel inteiro;

3) Parâmetros comuns nas simulações referentes à compressão de vídeo:

- Número de quadros por GOP: 12

### 6.3.2 Segundo Conjunto de Parâmetros

Esse segundo conjunto de parâmetros foi utilizado nas simulações com sequências em alta definição (HDTV) e também levou em consideração parâmetros recomendados pelo JVT, parâmetros utilizados com o objetivo de evidenciar as modificações introduzidas e parâmetros comuns nas simulações referentes à compressão de vídeo.

Foram realizadas 12 simulações com este conjunto de parâmetros, descritos a seguir:

1) Parâmetros recomendados pelo JVT:

- Resolução de vídeo: HDTV (1280x720);
- Codificação por Entropia: CABAC;
- Números de quadros usados como referência: 1;
- Hadamard: ligada
- Janela de Busca (W): 32

- *RDOptimization mode decision*: 1 (ligado no modo *High complexity*);
- Passos de Quantização (QP): 16, 20, 24 e 28 (incluindo o 1º quadro);
- Taxa de quadros: 60Hz (constante durante toda a seqüência);
- Estrutura do GOP: IBBP;
- Espaço de Cor: 4:2:0.

2) Parâmetros utilizados com o objetivo de evidenciar as modificações introduzidas:

- Seqüências de vídeo: Utilizar seqüências que possuam cenas com médio e baixo movimento. Dessa maneira escolhemos a seqüência *Stockholm*;
- Algoritmos: Utilizar os algoritmos implementados no código do H.264/AVC. Dessa forma foram escolhidos os algoritmos FSA e UMHS para as análises em conjunto com o AUMHS;
- Acuracidade do Pixel: Pixel inteiro e fracionário. As modificações introduzidas estão na parte do código para pixel inteiro, porém neste conjunto de simulações habilitamos a opção de pixel fracionário para demonstrar a efetividade do AUMHS, mesmo com esta opção habilitada;

3) Parâmetros comuns nas simulações referentes à compressão de vídeo:

- Número de quadros por GOP: 12

## 6.4

### Resultados

#### 6.4.1

##### Primeira Seção dos Resultados

Nesta seção apresentamos os resultados alcançados com o algoritmo AUMHS (utilizando o primeiro conjunto de parâmetros), em comparação aos algoritmos FSA e UMHS, em termos de complexidade computacional e qualidade.

O item complexidade computacional é avaliado através das seguintes medidas:

- Tempo total de processamento do codificador em segundos (TT);
- Tempo de processamento do módulo de estimação de movimento em segundos (TEM).

O item qualidade é avaliado através das seguintes medidas:

- *Peak Signal to Noise Ratio* da componente de luminância em decibéis (PSNR-Y);
- *Peak Signal to Noise Ratio* da 1ª componente de croma em decibéis (PSNR-U);
- *Peak Signal to Noise Ratio* da 2ª componente de croma em decibéis (PSNR-V);
- Taxa de bits utilizada em Kbps (TB);
- Avaliação subjetiva através de amostras de quadros das seqüências codificadas.

Os resultados apresentados estão divididos de acordo com as seqüências de vídeo utilizadas. Logo, as tabelas com os resultados das simulações, para cada seqüência de vídeo, assim como as conclusões observadas, são mostradas sempre na seguinte ordem:

- Tabela com os resultados do FSA;
- Tabela com os resultados do UMHS;
- Tabela com os resultados do AUMHS;
- Tabela com a análise comparativa entre FSA x AUMHS;
- Tabela com a análise comparativa entre UMHS x AUMHS;
- Amostra de quadro de imagem da seqüência codificada com cada um dos três algoritmos em questão;
- Conclusão obtida através da análise comparativa.

### Resultados dos Algoritmos FSA, UMHS e AUMHS utilizando a sequência QCIF *walk*

A Tabela 6.1 traz os primeiros resultados das simulações realizadas com o algoritmo FSA, onde utilizou-se a sequência *walk*, considerada uma sequência com cenas que apresentam alto movimento, variando os passos de quantização:

Tabela 6.1: Resultado das simulações com o algoritmo FSA utilizando a sequência *walk*.

| Nº | QP | PSNR-Y<br>(dB) | PSNR-U<br>(dB) | PSNR-V<br>(dB) | TB<br>(Kbps) | TT<br>(s) | TEM<br>(s) |
|----|----|----------------|----------------|----------------|--------------|-----------|------------|
| 1  | 16 | 44,8           | 46,6           | 45,6           | 1746,3       | 208,8     | 99,0       |
| 2  | 20 | 41,3           | 44,2           | 43,2           | 1185,2       | 226,8     | 117,3      |
| 3  | 24 | 38,0           | 41,9           | 41,1           | 786,2        | 178,3     | 99,8       |
| 4  | 28 | 38,0           | 41,9           | 41,1           | 786,2        | 178,5     | 100,4      |

A Tabela 6.2 apresenta os resultados das simulações realizadas com o algoritmo UMHS:

Tabela 6.2: Resultado das simulações com o algoritmo UMHS utilizando a sequência *walk*.

| Nº | QP | PSNR-Y<br>(dB) | PSNR-U<br>(dB) | PSNR-V<br>(dB) | TB<br>(Kbps) | TT<br>(s) | TEM<br>(s) |
|----|----|----------------|----------------|----------------|--------------|-----------|------------|
| 1  | 16 | 44,8           | 46,6           | 45,6           | 1747,9       | 122,5     | 11,8       |
| 2  | 20 | 41,3           | 44,2           | 43,2           | 1186,9       | 102,1     | 11,6       |
| 3  | 24 | 38,1           | 41,9           | 41,1           | 787,8        | 89,1      | 11,9       |
| 4  | 28 | 38,1           | 41,9           | 41,1           | 787,8        | 89,2      | 12,0       |

A Tabela 6.3 apresenta os resultados das simulações realizadas com o algoritmo AUMHS:

A Tabela 6.4 apresenta a análise comparativa entre os algoritmos FSA e AUMHS:

A Tabela 6.5 apresenta a análise comparativa entre os algoritmos UMHS e AUMHS.

A Figura 6.1 apresenta uma amostra de um quadro de imagem codificado com cada um dos algoritmos FSA, UMHS e AUMHS. Neste exemplo utilizamos a simulação com passo de quantização igual a 16.

Através das avaliações apresentadas nas tabelas e nas imagens acima concluímos que com relação ao algoritmo FSA a qualidade média obtida,

Tabela 6.3: Resultado das simulações com o algoritmo AUMHS utilizando a seqüência *walk*.

| Nº | QP | PSNR-Y<br>(dB) | PSNR-U<br>(dB) | PSNR-V<br>(dB) | TB<br>(Kbps) | TT<br>(s) | TEM<br>(s) |
|----|----|----------------|----------------|----------------|--------------|-----------|------------|
| 1  | 16 | 44,8           | 46,5           | 45,6           | 1755,4       | 114,0     | 8,2        |
| 2  | 20 | 41,3           | 44,2           | 43,2           | 1198,5       | 93,2      | 5,4        |
| 3  | 24 | 38,0           | 41,9           | 41,1           | 795,5        | 83,2      | 7,3        |
| 4  | 28 | 38,0           | 41,9           | 41,1           | 795,5        | 76,5      | 8,9        |

Tabela 6.4: Análise comparativa entre os algoritmos FSA e AUMHS para a seqüência *walk*.

| Nº | QP | Economia<br>de TT | Economia<br>de TEM | Ganho de<br>PSNR-Y | Aumento<br>de TB |
|----|----|-------------------|--------------------|--------------------|------------------|
| 1  | 16 | 45,4%             | 91,7%              | 0,0%               | 0,5%             |
| 2  | 20 | 58,9%             | 95,4%              | 0,0%               | 1,1%             |
| 3  | 24 | 53,3%             | 92,7%              | 0,0%               | 1,2%             |
| 4  | 28 | 45,9%             | 91,2%              | 0,0%               | 1,2%             |

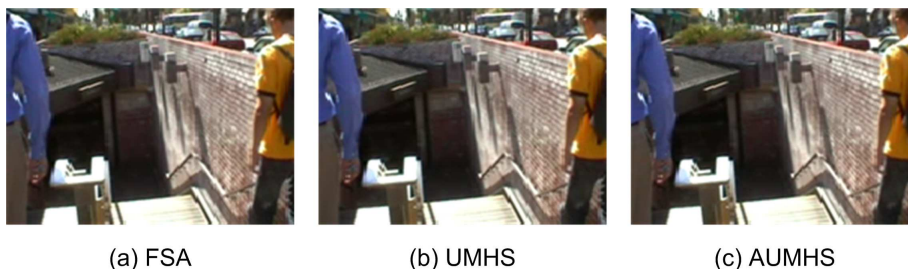


Figura 6.1: Quadro de imagem, da seqüência *walk*, codificado com os algoritmos FSA, UMHS e AUMHS.

em termos de PSNR-Y, foi exatamente a mesma com apenas um acréscimo médio de 1% da taxa de bits utilizada. No mesmo item de qualidade, a comparação com o UMHS atingiu os mesmos resultados.

Ainda assim conseguiu-se uma redução média da complexidade computacional da ordem de 54%, com relação ao FSA, e de 9% com relação ao UMHS no que se diz respeito ao tempo total de processamento do codificador. E com relação ao tempo de processamento do módulo de estimção de movimento os resultados na média alcançaram 93% e 37% quando comparado ao FSA e ao UMHS respectivamente.

Tal resultado se deve ao fato de que apesar da seqüência *walk* possuir várias cenas com alto teor de movimento, ainda assim existe um grande número de cenas com médio e baixo movimento que permitem a otimização

Tabela 6.5: Análise comparativa entre os algoritmos UMHS e AUMHS para a seqüência *walk*.

| Nº | QP | Economia<br>de TT | Economia<br>de TEM | Ganho de<br>PSNR-Y | Aumento<br>de TB |
|----|----|-------------------|--------------------|--------------------|------------------|
| 1  | 16 | 7,0%              | 30,5%              | 0,0%               | 0,4%             |
| 2  | 20 | 8,7%              | 53,7%              | 0,0%               | 1,0%             |
| 3  | 24 | 6,6%              | 38,7%              | 0,0%               | 1,0%             |
| 4  | 28 | 8,1%              | 26,1%              | 0,0%               | 1,0%             |

dos parâmetros a serem utilizados para a estimção de movimento e para o restante da codificação.

### Resultados dos Algoritmos FSA, UMHS e AUMHS utilizando a seqüência *QCIF foreman*

A Tabela 6.6 traz os resultados das simulações realizadas com o algoritmo FSA, onde utilizou-se a seqüência *foreman*, considerada uma seqüência com cenas que apresentam médio movimento, variando os passos de quantização:

Tabela 6.6: Resultado das simulações com o algoritmo FSA utilizando a seqüência *foreman*.

| Nº | QP | PSNR-Y<br>(dB) | PSNR-U<br>(dB) | PSNR-V<br>(dB) | TB<br>(Kbps) | TT<br>(s) | TEM<br>(s) |
|----|----|----------------|----------------|----------------|--------------|-----------|------------|
| 5  | 16 | 44,5           | 45,6           | 47,2           | 1151,8       | 199,1     | 97,1       |
| 6  | 20 | 41,0           | 43,2           | 45,4           | 664,1        | 181,5     | 97,3       |
| 7  | 24 | 38,0           | 41,1           | 43,1           | 365,8        | 169,0     | 97,9       |
| 8  | 28 | 35,2           | 39,1           | 40,9           | 191,4        | 160,0     | 98,4       |

A Tabela 6.7 apresenta os resultados das simulações realizadas com o algoritmo UMHS:

A Tabela 6.8 apresenta os resultados das simulações realizadas com o algoritmo AUMHS:

A Tabela 6.9 apresenta a análise comparativa entre os algoritmos FSA e AUMHS.

A Tabela 6.10 apresenta a análise comparativa entre os algoritmos UMHS e AUMHS.



Tabela 6.7: Resultado das simulações com o algoritmo UMHS utilizando a sequência *foreman*.

| Nº | QP | PSNR-Y<br>(dB) | PSNR-U<br>(dB) | PSNR-V<br>(dB) | TB<br>(Kbps) | TT<br>(s) | TEM<br>(s) |
|----|----|----------------|----------------|----------------|--------------|-----------|------------|
| 5  | 16 | 44,5           | 45,6           | 47,2           | 1159,0       | 106,0     | 5,3        |
| 6  | 20 | 41,0           | 43,2           | 45,4           | 672,3        | 88,5      | 5,4        |
| 7  | 24 | 37,9           | 41,1           | 43,1           | 370,1        | 89,7      | 6,3        |
| 8  | 28 | 35,2           | 39,1           | 40,9           | 191,7        | 66,2      | 5,8        |

Tabela 6.8: Resultado das simulações com o algoritmo AUMHS utilizando a sequência *foreman*.

| Nº | QP | PSNR-Y<br>(dB) | PSNR-U<br>(dB) | PSNR-V<br>(dB) | TB<br>(Kbps) | TT<br>(s) | TEM<br>(s) |
|----|----|----------------|----------------|----------------|--------------|-----------|------------|
| 5  | 16 | 44,5           | 45,6           | 47,2           | 1179,5       | 102,9     | 4,4        |
| 6  | 20 | 41,0           | 43,2           | 45,4           | 684,8        | 86,2      | 4,4        |
| 7  | 24 | 37,9           | 41,1           | 43,1           | 376,8        | 74,1      | 4,5        |
| 8  | 28 | 35,2           | 39,2           | 40,9           | 195,5        | 64,4      | 4,9        |

A Figura 6.2 apresenta uma amostra de um quadro de imagem codificado com cada um dos algoritmos FSA, UMHS e AUMHS. Neste exemplo utilizamos a simulação com passo de quantização igual a 20.

Figura 6.2: Quadro de imagem, da sequência *foreman*, codificado com os algoritmos FSA, UMHS e AUMHS.

Através das avaliações apresentadas nas tabelas e nas imagens acima concluímos que com relação ao algoritmo FSA a qualidade média obtida, em termos de PSNR-Y, foi exatamente a mesma com apenas um acréscimo médio de 3% da taxa de bits utilizada. No mesmo item de qualidade, a comparação com o UMHS atingiu os mesmos resultados em termos de PSNR-Y, porém com um acréscimo menor da taxa de bits utilizada, 2%.

Ainda assim conseguiu-se uma redução média da complexidade computacional da ordem de 54%, com relação ao FSA, e de 6% com relação ao

Tabela 6.9: Análise comparativa entre os algoritmos FSA e AUMHS para a sequência *foreman*.

| Nº | QP | Economia de TT | Economia de TEM | Ganho de PSNR-Y | Aumento de TB |
|----|----|----------------|-----------------|-----------------|---------------|
| 5  | 16 | 48,3%          | 95,5%           | -0,1%           | 2,3%          |
| 6  | 20 | 52,5%          | 95,5%           | -0,1%           | 3,0%          |
| 7  | 24 | 56,2%          | 95,4%           | -0,2%           | 2,9%          |
| 8  | 28 | 59,8%          | 95,1%           | -0,2%           | 2,1%          |

Tabela 6.10: Análise comparativa entre os algoritmos UMHS e AUMHS para a sequência *foreman*.

| Nº | QP | Economia de TT | Economia de TEM | Ganho de PSNR-Y | Aumento de TB |
|----|----|----------------|-----------------|-----------------|---------------|
| 5  | 16 | 2,9%           | 17,5%           | -0,1%           | 1,7%          |
| 6  | 20 | 2,6%           | 18,4%           | -0,1%           | 1,8%          |
| 7  | 24 | 17,4%          | 27,7%           | -0,1%           | 1,8%          |
| 8  | 28 | 2,7%           | 15,3%           | -0,1%           | 2,0%          |

UMHS no que se diz respeito ao tempo total de processamento do codificador. E com relação ao tempo de processamento do módulo de estimação de movimento os resultados médios alcançaram 95% e 20% quando comparado ao FSA e ao UMHS respectivamente.

### Resultados dos Algoritmos FSA, UMHS e AUMHS utilizando a sequência *QCIF container*

A Tabela 6.11 traz os resultados das simulações realizadas com o algoritmo FSA, onde utilizou-se a sequência *container*, considerada uma sequência com cenas que apresentam baixo movimento, variando os passos de quantização.

A Tabela 6.12 apresenta os resultados das simulações realizadas com o algoritmo UMHS.

A Tabela 6.13 apresenta os resultados das simulações realizadas com o algoritmo AUMHS.

A Tabela 6.14 apresenta a análise comparativa entre os algoritmos FSA e AUMHS.

A Tabela 6.15 apresenta a análise comparativa entre os algoritmos UMHS e AUMHS.

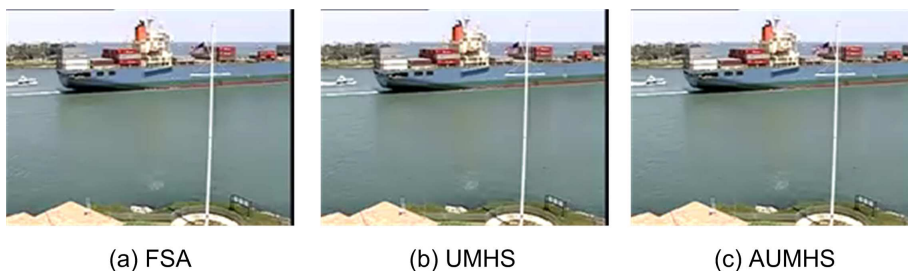
Tabela 6.11: Resultado das simulações com o algoritmo FSA utilizando a sequência *container*.

| Nº | QP | PSNR-Y<br>(dB) | PSNR-U<br>(dB) | PSNR-V<br>(dB) | TB<br>(Kbps) | TT<br>(s) | TEM<br>(s) |
|----|----|----------------|----------------|----------------|--------------|-----------|------------|
| 9  | 16 | 45,2           | 48,3           | 48,3           | 466,6        | 188,6     | 97,3       |
| 10 | 20 | 42,2           | 46,1           | 46,0           | 286,0        | 175,2     | 97,8       |
| 11 | 24 | 39,3           | 43,4           | 43,5           | 175,3        | 165,5     | 97,9       |
| 12 | 28 | 36,6           | 41,3           | 40,9           | 107,7        | 167,9     | 104,5      |

Tabela 6.12: Resultado das simulações com o algoritmo UMHS utilizando a sequência *container*.

| Nº | QP | PSNR-Y<br>(dB) | PSNR-U<br>(dB) | PSNR-V<br>(dB) | TB<br>(Kbps) | TT<br>(s) | TEM<br>(s) |
|----|----|----------------|----------------|----------------|--------------|-----------|------------|
| 9  | 16 | 45,2           | 48,3           | 48,3           | 466,4        | 93,0      | 2,6        |
| 10 | 20 | 42,2           | 46,1           | 46,0           | 286,0        | 78,8      | 2,7        |
| 11 | 24 | 39,3           | 43,4           | 43,5           | 175,3        | 69,0      | 3,1        |
| 12 | 28 | 36,6           | 41,3           | 40,9           | 107,9        | 71,6      | 4,4        |

A Figura 6.3 apresenta uma amostra de um quadro de imagem codificado com cada um dos algoritmos FSA, UMHS e AUMHS. Neste exemplo utilizamos a simulação com passo de quantização igual a 28.

Figura 6.3: Quadro de imagem, da sequência *container*, codificado com os algoritmos FSA, UMHS e AUMHS.

Através das avaliações apresentadas nas tabelas e nas imagens acima concluímos que tanto com relação ao algoritmo FSA quando com relação ao UMHS a qualidade média obtida, em termos de PSNR-Y, foi exatamente a mesma e sem nenhum acréscimo na taxa de bits utilizada.

Ainda assim conseguiu-se uma redução média da complexidade computacional da ordem de 57%, com relação ao FSA, e de 5% com relação ao UMHS no que se diz respeito ao tempo total de processamento do codificador. E com relação ao tempo de processamento do módulo de estimação de

Tabela 6.13: Resultado das simulações com o algoritmo AUMHS utilizando a sequência *container*.

| Nº | QP | PSNR-Y<br>(dB) | PSNR-U<br>(dB) | PSNR-V<br>(dB) | TB<br>(Kbps) | TT<br>(s) | TEM<br>(s) |
|----|----|----------------|----------------|----------------|--------------|-----------|------------|
| 9  | 16 | 45,2           | 48,3           | 48,3           | 466,2        | 88,7      | 1,1        |
| 10 | 20 | 42,1           | 46,1           | 46,0           | 284,4        | 75,2      | 1,0        |
| 11 | 24 | 39,3           | 43,4           | 43,5           | 175,2        | 65,0      | 1,4        |
| 12 | 28 | 36,6           | 41,3           | 40,9           | 108,0        | 69,3      | 1,5        |

Tabela 6.14: Análise comparativa entre os algoritmos FSA e AUMHS para a sequência *container*.

| Nº | QP | Economia<br>de TT | Economia<br>de TEM | Ganho de<br>PSNR-Y | Aumento<br>de TB |
|----|----|-------------------|--------------------|--------------------|------------------|
| 9  | 16 | 53,0%             | 98,9%              | 0,0%               | -0,1%            |
| 10 | 20 | 57,1%             | 99,0%              | 0,0%               | -0,6%            |
| 11 | 24 | 60,7%             | 98,6%              | 0,0%               | -0,1%            |
| 12 | 28 | 58,8%             | 98,6%              | 0,0%               | 0,2%             |

movimento os resultados médios alcançaram 99% e 61% quando comparado ao FSA e ao UMHS respectivamente.

#### 6.4.2

##### Segunda Seção dos Resultados

Nesta seção apresentamos os resultados alcançados com o algoritmo AUMHS (utilizando o primeiro e o segundo conjunto de parâmetros), em comparação aos algoritmos FSA e UMHS, em termos de complexidade computacional e qualidade.

Os itens de complexidade computacional e qualidade são avaliados da mesma maneira realizada com as simulações apresentadas anteriormente, assim como também repetimos o método de apresentação dos resultados.

##### Resultados dos Algoritmos FSA, UMHS e AUMHS utilizando a sequência HDTV *stockholm* e o primeiro conjunto de parâmetros de simulação

A Tabela 6.16 traz os resultados das simulações realizadas com o algoritmo FSA, onde utilizou-se a sequência *stockholm*, considerada uma

Tabela 6.15: Análise comparativa entre os algoritmos UMHS e AUMHS para a sequência *container*.

| Nº | QP | Economia de TT | Economia de TEM | Ganho de PSNR-Y | Aumento de TB |
|----|----|----------------|-----------------|-----------------|---------------|
| 9  | 16 | 4,7%           | 59,4%           | 0,0%            | -0,1%         |
| 10 | 20 | 4,5%           | 62,7%           | 0,0%            | -0,6%         |
| 11 | 24 | 5,8%           | 56,1%           | 0,0%            | -0,1%         |
| 12 | 28 | 3,2%           | 65,9%           | 0,0%            | 0,0%          |

sequência com cenas que apresentam médio e baixo movimentos, variando os passos de quantização:

Tabela 6.16: Resultado das simulações com o algoritmo FSA utilizando a sequência *stockholm* e o primeiro conjunto de parâmetros de simulação.

| Nº | QP | PSNR-Y (dB) | PSNR-U (dB) | PSNR-V (dB) | TB (Kbps) | TT (s) | TEM (s) |
|----|----|-------------|-------------|-------------|-----------|--------|---------|
| 1  | 16 | 47,8        | 52,9        | 53,1        | 32608,2   | 1014,4 | 500,2   |
| 2  | 20 | 44,1        | 51,1        | 51,2        | 22372,3   | 899,0  | 486,2   |
| 3  | 24 | 41,2        | 48,0        | 47,9        | 12404,9   | 945,9  | 536,0   |
| 4  | 28 | 38,6        | 45,7        | 45,4        | 6942,6    | 815,4  | 488,3   |

A Tabela 6.17 apresenta os resultados das simulações realizadas com o algoritmo UMHS e o primeiro conjunto de parâmetros de simulação:

A Tabela 6.18 apresenta os resultados das simulações realizadas com o algoritmo AUMHS e o primeiro conjunto de parâmetros de simulação:

A Tabela 6.19 apresenta a análise comparativa entre os algoritmos FSA e AUMHS para o primeiro conjunto de parâmetros de simulação:

A Tabela 6.20 apresenta a análise comparativa entre os algoritmos UMHS e AUMHS para o primeiro conjunto de parâmetros de simulação.

As Figuras 6.4 6.5 6.6 apresentam uma amostra de um quadro de imagem codificado com o primeiro conjunto de parâmetros para cada um dos algoritmos FSA, UMHS e AUMHS. Nestes exemplos utilizamos a simulação com passo de quantização igual a 16.

Através das avaliações apresentadas nas tabelas e nas imagens acima concluímos que com relação ao algoritmo FSA a qualidade média obtida, em termos de PSNR-Y, foi praticamente a mesma com apenas um acréscimo médio de 2% da taxa de bits utilizada. No mesmo item de qualidade, a comparação com o UMHS também atingiu resultados satisfatórios.

Tabela 6.17: Resultado das simulações com o algoritmo UMHS utilizando a sequência *stockholm* e o primeiro conjunto de parâmetros de simulação.

| Nº | QP | PSNR-Y<br>(dB) | PSNR-U<br>(dB) | PSNR-V<br>(dB) | TB<br>(Kbps) | TT<br>(s) | TEM<br>(s) |
|----|----|----------------|----------------|----------------|--------------|-----------|------------|
| 1  | 16 | 47,8           | 52,9           | 53,1           | 33260,2      | 515,0     | 5,3        |
| 2  | 20 | 44,1           | 51,1           | 51,2           | 22491,8      | 421,3     | 13,3       |
| 3  | 24 | 41,2           | 48,0           | 47,9           | 12459,0      | 391,2     | 13,1       |
| 4  | 28 | 38,6           | 45,7           | 45,4           | 6980,4       | 342,4     | 12,5       |

Tabela 6.18: Resultado das simulações com o algoritmo AUMHS utilizando a sequência *stockholm* e o primeiro conjunto de parâmetros de simulação.

| Nº | QP | PSNR-Y<br>(dB) | PSNR-U<br>(dB) | PSNR-V<br>(dB) | TB<br>(Kbps) | TT<br>(s) | TEM<br>(s) |
|----|----|----------------|----------------|----------------|--------------|-----------|------------|
| 1  | 16 | 47,8           | 52,9           | 53,1           | 33660,2      | 480,4     | 4,9        |
| 2  | 20 | 44,0           | 51,0           | 51,1           | 22704,5      | 399,0     | 5,2        |
| 3  | 24 | 41,1           | 48,0           | 47,9           | 12582,3      | 351,5     | 4,9        |
| 4  | 28 | 38,6           | 45,6           | 45,4           | 7073,8       | 317,1     | 5,2        |

Ainda assim conseguiu-se uma redução média da complexidade computacional da ordem de 58%, com relação ao FSA, e de 7,4% com relação ao UMHS no que se diz respeito ao tempo total de processamento do codificador. E com relação ao tempo de processamento do módulo de estimção de movimento os resultados na média alcançaram 96,8% e 47,4% quando comparado ao FSA e ao UMHS respectivamente.

### Resultados dos Algoritmos FSA, UMHS e AUMHS utilizando a sequência HDTV *stockholm* e o segundo conjunto de parâmetros de simulação

A Tabela 6.21 traz os resultados das simulações realizadas com o algoritmo FSA, onde utilizou-se a sequência *stockholm*, considerada uma sequência com cenas que apresentam médio e baixo movimentos, variando os passos de quantização:

A Tabela 6.22 apresenta os resultados das simulações realizadas com o algoritmo UMHS e o segundo conjunto de parâmetros de simulação:

A Tabela 6.23 apresenta os resultados das simulações realizadas com o algoritmo AUMHS e o segundo conjunto de parâmetros de simulação:

A Tabela 6.24 apresenta a análise comparativa entre os algoritmos FSA e AUMHS para o segundo conjunto de parâmetros de simulação:

Tabela 6.19: Análise comparativa entre os algoritmos FSA e AUMHS para a sequência *stockholm* e o primeiro conjunto de parâmetros de simulação.

| Nº | QP | Economia de TT | Economia de TEM | Ganho de PSNR-Y | Aumento de TB |
|----|----|----------------|-----------------|-----------------|---------------|
| 1  | 16 | 52,6%          | 90,2%           | 0,0%            | 3,2%          |
| 2  | 20 | 55,6%          | 98,9%           | 0,0%            | 1,5%          |
| 3  | 24 | 62,8%          | 99,1%           | -0,2%           | 1,4%          |
| 4  | 28 | 61,1%          | 98,9%           | 0,0%            | 1,9%          |

Tabela 6.20: Análise comparativa entre os algoritmos UMHS e AUMHS para a sequência *stockholm* e o primeiro conjunto de parâmetros de simulação.

| Nº | QP | Economia de TT | Economia de TEM | Ganho de PSNR-Y | Aumento de TB |
|----|----|----------------|-----------------|-----------------|---------------|
| 1  | 16 | 6,7%           | 7,6%            | 0,0%            | 1,2%          |
| 2  | 20 | 5,3%           | 60,9%           | -0,2%           | 1,0%          |
| 3  | 24 | 10,2%          | 62,6%           | -0,2%           | 1,0%          |
| 4  | 28 | 7,4%           | 58,4%           | 0,0%            | 1,3%          |

A Tabela 6.25 apresenta a análise comparativa entre os algoritmos UMHS e AUMHS para o segundo conjunto de parâmetros de simulação.

As Figuras 6.7 6.8 6.9 apresentam uma amostra de um quadro de imagem codificado com o segundo conjunto de parâmetros para cada um dos algoritmos FSA, UMHS e AUMHS. Nestes exemplos utilizamos a simulação com passo de quantização igual a 20.

Através das avaliações apresentadas nas tabelas e nas imagens acima concluímos que com relação ao algoritmo FSA a qualidade média obtida, em termos de PSNR-Y, foi praticamente a mesma (pequeno decréscimo de 0,2%) com apenas um acréscimo médio de 1% da taxa de bits utilizada. Da mesma forma, a comparação com o UMHS atingiu os resultados de decréscimo de 0,2% em termos de PSNR-Y e acréscimo de 0,8% em termos da taxa de bits utilizada.

Ainda assim conseguiu-se uma redução média da complexidade computacional da ordem de 56,3%, com relação ao FSA, e de 13,8% com relação ao UMHS no que se diz respeito ao tempo total de processamento do codificador. E com relação ao tempo de processamento do módulo de estimção de movimento os resultados na média alcançaram 90,9% e 34,1% quando comparado ao FSA e ao UMHS respectivamente.

Através das 60 simulações e das análises comparativas realizadas



Figura 6.4: Quadro de imagem, da sequência *stockholm*, codificado com o algoritmo FSA.

Tabela 6.21: Resultado das simulações com o algoritmo FSA utilizando a sequência *stockholm* e o segundo conjunto de parâmetros de simulação.

| Nº | QP | PSNR-Y<br>(dB) | PSNR-U<br>(dB) | PSNR-V<br>(dB) | TB<br>(Kbps) | TT<br>(s) | TEM<br>(s) |
|----|----|----------------|----------------|----------------|--------------|-----------|------------|
| 1  | 16 | 48,4           | 53,2           | 53,4           | 28680,5      | 1041,1    | 554,9      |
| 2  | 20 | 44,7           | 51,4           | 51,5           | 16936,8      | 958,9     | 552,7      |
| 3  | 24 | 42,0           | 48,3           | 48,2           | 9608,2       | 912,2     | 552,7      |
| 4  | 28 | 39,6           | 45,9           | 45,6           | 5907,7       | 961,5     | 603,6      |

neste capítulo foi possível observar que o algoritmo AUMHS proposto consegue conservar a qualidade da imagem em termos subjetivos e objetivos, alcançando significativa redução da velocidade de processamento quando comparado aos algoritmos FSA e UMHS. Estas 60 simulações levaram em consideração sequências de vídeo com resolução QCIF ou HDTV e a variação dos passos de quantização e da acuracidade do pixel.

Por utilizar uma técnica de medida de movimento, este algoritmo apresenta excelentes resultados, principalmente em cenas com baixo movimento, possibilitando a utilização eficiente dos parâmetros dos algoritmos de estimação de movimento e dos parâmetros de codificação de uma maneira geral.





Figura 6.5: Quadro de imagem, da sequência *stockholm*, codificado com o algoritmo UMHS.

Tabela 6.22: Resultado das simulações com o algoritmo UMHS utilizando a sequência *stockholm* e o segundo conjunto de parâmetros de simulação.

| Nº | QP | PSNR-Y<br>(dB) | PSNR-U<br>(dB) | PSNR-V<br>(dB) | TB<br>(Kbps) | TT<br>(s) | TEM<br>(s) |
|----|----|----------------|----------------|----------------|--------------|-----------|------------|
| 1  | 16 | 48,4           | 53,2           | 53,4           | 28712,3      | 561,3     | 74,2       |
| 2  | 20 | 44,7           | 51,4           | 51,5           | 16958,2      | 528,4     | 83,4       |
| 3  | 24 | 42,0           | 48,3           | 48,2           | 9645,9       | 487,2     | 84,7       |
| 4  | 28 | 39,6           | 45,9           | 45,6           | 5919,4       | 397,6     | 72,5       |



Figura 6.6: Quadro de imagem, da sequência *stockholm*, codificado com o algoritmo AUMHS.

Tabela 6.23: Resultado das simulações com o algoritmo AUMHS utilizando a seqüência *stockholm* e o segundo conjunto de parâmetros de simulação.

| Nº | QP | PSNR-Y<br>(dB) | PSNR-U<br>(dB) | PSNR-V<br>(dB) | TB<br>(Kbps) | TT<br>(s) | TEM<br>(s) |
|----|----|----------------|----------------|----------------|--------------|-----------|------------|
| 1  | 16 | 48,3           | 53,2           | 53,3           | 29228,8      | 501,9     | 50,7       |
| 2  | 20 | 44,6           | 51,4           | 51,5           | 17138,3      | 442,5     | 52,2       |
| 3  | 24 | 41,9           | 48,3           | 48,2           | 9661,4       | 396,0     | 52,9       |
| 4  | 28 | 39,5           | 45,9           | 45,6           | 5924,0       | 359,1     | 50,9       |

Tabela 6.24: Análise comparativa entre os algoritmos FSA e AUMHS para a seqüência *stockholm* e o segundo conjunto de parâmetros de simulação.

| Nº | QP | Economia<br>de TT | Economia<br>de TEM | Ganho de<br>PSNR-Y | Aumento<br>de TB |
|----|----|-------------------|--------------------|--------------------|------------------|
| 1  | 16 | 51,8%             | 90,9%              | -0,2%              | 1,9%             |
| 2  | 20 | 53,9%             | 90,6%              | -0,2%              | 1,2%             |
| 3  | 24 | 56,6%             | 90,4%              | -0,2%              | 0,6%             |
| 4  | 28 | 62,7%             | 91,6%              | -0,3%              | 0,3%             |

Tabela 6.25: Análise comparativa entre os algoritmos UMHS e AUMHS para a seqüência *stockholm* e o segundo conjunto de parâmetros de simulação.

| Nº | QP | Economia<br>de TT | Economia<br>de TEM | Ganho de<br>PSNR-Y | Aumento<br>de TB |
|----|----|-------------------|--------------------|--------------------|------------------|
| 1  | 16 | 10,6%             | 31,7%              | -0,2%              | 1,8%             |
| 2  | 20 | 16,3%             | 37,4%              | -0,2%              | 1,1%             |
| 3  | 24 | 18,7%             | 37,5%              | -0,2%              | 0,2%             |
| 4  | 28 | 9,7%              | 29,8%              | -0,3%              | 0,1%             |



Figura 6.7: Quadro de imagem, da sequência *stockholm*, codificado com o algoritmo FSA.



Figura 6.8: Quadro de imagem, da sequência *stockholm*, codificado com o algoritmo UMHS.



Figura 6.9: Quadro de imagem, da sequência *stockholm*, codificado com o algoritmo AUMHS.