

6

Escolha da Direção Preferencial de Processamento da SA-DCT

6.1

Introdução

De forma semelhante ao que foi feito no Capítulo 5 para a EI-DCT, este capítulo tratará da escolha da direção preferencial de processamento da SA-DCT, que constitui a outra abordagem de transformadas orientadas por objeto examinadas nesta tese. Após a análise de algoritmos apresentados na literatura, será proposta uma estrutura adaptativa híbrida para a escolha da direção do primeiro conjunto de transformadas 1D da SA-DCT [67]-[75].

O algoritmo original da SA-DCT [67] aplica o primeiro conjunto de transformadas 1D sobre os segmentos na direção vertical. Estudos posteriores concentraram-se nos seguintes tópicos: (i) análise, formulação matemática e estratégias de alinhamento dos coeficientes verticais da transformada antes do processamento das transformadas 1D na direção horizontal [69]-[75] e (ii) escolha da direção preferencial para o processamento do primeiro conjunto de transformadas 1D [77]. Este capítulo, portanto, abordará o segundo tema.

Em [77], o critério que escolhe como prioritária a direção que produz o menor produto $emq \times nbits$ também foi apresentado como uma opção eficiente para a escolha da direção preferencial de processamento para a SA-DCT, onde emq é o erro médio quadrático e $nbits$ é o número de bits empregados na quantização. Como ocorre para o algoritmo de extrapolação EI, também na SA-DCT esse algoritmo supera as outras estratégias para a determinação da direção preferencial, mas apresenta a desvantagem de produzir um conjunto de coeficientes distinto para cada taxa de bits.

As estratégias baseadas nas variâncias dos comprimentos dos segmentos do objeto, MILV e MALV [77], também são aplicáveis à SA-DCT. Neste capítulo, esses esquemas serão analisados sob a ótica dos indicadores morfológicos apresentados na Seção 4.3, assim como foi feito para a EI-DCT no

Capítulo 5. Em seguida, será proposto um esquema híbrido que será capaz de fazer uma seleção adaptativa da estratégia mais adequada a cada grupo específico de blocos de contorno, de acordo com o indicador TNPO.

A Seção 6.2 apresenta o algoritmo SA-DCT e alguns resultados relativos à análise preliminar do desempenho de MILV e MALV quando aplicados sobre os grupos de blocos de contorno. Esses resultados irão subsidiar a proposição do esquema híbrido na Seção 6.3, onde também serão apresentados resultados de simulação e discussões. Na Seção 6.4, serão avaliados os esquemas de particionamento da imagem em conjunto com o novo método proposto e, finalmente, na Seção 6.5, serão destacadas as principais contribuições deste capítulo.

6.2

O algoritmo SA-DCT e Alguns Resultados Preliminares

Na SA-DCT, os blocos de contorno de dimensão $N \times N$ são transformados em segmentos verticais de comprimento variável L ($1 \leq L \leq N$), que contêm somente pixels que pertençam ao objeto. Em seguida são aplicadas transformadas DCT-1D definidas por $S_L.[DCT - L]$, onde a matriz $[DCT - L]$ é dada por:

$$[DCT - L] = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \cdots & \frac{1}{\sqrt{2}} \\ \cos \frac{\pi}{2L} & \cos \frac{3\pi}{2L} & \cdots & \cos \frac{(2L-1)\pi}{2L} \\ \vdots & \vdots & & \vdots \\ \cos \frac{(L-1)\pi}{2L} & \cos \frac{3(L-1)\pi}{2L} & \cdots & \cos \frac{(2L-1)(L-1)\pi}{2L} \end{bmatrix} \quad (6-1)$$

Em um segundo estágio, os coeficientes verticais resultantes são alinhados de acordo com o seu índice (ou ordem), e processados na direção horizontal usando novamente a matriz de comprimento variável $S_L.[DCT - L]$ [67]. No caso do algoritmo de Sikora, faz-se $S_L = 2/L$, de forma que $S_L L$ seja constante. Isso garante que o valor médio da imagem seja completamente mapeado no coeficiente DC.

Como mencionado no Capítulo 2, ao invés de aplicar as transformadas $S_L.[DCT - L]$ aos segmentos verticais primeiro — o que corresponde à estratégia ‘Columns-First’ — a direção preferencial de processamento pode ser escolhida de forma adaptativa com o objetivo de alcançar uma melhor eficiência de codificação. Em [77], os autores compararam o critério que

escolhe a direção que produz o menor produto $emq \times nbits$ às estratégias ‘Columns-First’, ‘Rows-First’, MALV e MILV. Com base apenas nos resultados de codificação de *AKYIO*, eles concluíram que MALV é preferível a MILV. Entretanto, nós observamos, a partir de alguns resultados preliminares, que, assim como na EI-DCT, também com o algoritmo SA-DCT alguns objetos beneficiam-se da estratégia MALV, ao passo que outros tiram maior proveito da estratégia MILV.

Verificou-se ainda que o indicador morfológico TNPO, que incorpora características de forma e de textura do objeto, também pode ser usado para analisar as estratégias para a determinação da direção preferencial de processamento da SA-DCT. Assim como ocorreu no algoritmo EI, também se observou que a eficiência das estratégias MILV e MALV apresentam variações não somente de objeto para objeto, mas também de grupo para grupo de blocos de contorno definidos no indicador TNPO. Isso significa que uma análise cuidadosa desse indicador pode ser útil para a seleção da estratégia mais adequada para ser usada em cada um dos grupos de blocos.

A estratégia que prioriza a direção com a maior variância nos comprimentos - MALV - irá processar primeiro os segmentos com comprimentos menos uniformes (isso significa que alguns segmentos serão significativamente mais longos, enquanto outros serão muito pequenos). Por outro lado, a estratégia que escolhe a direção com a menor variância dos comprimentos - MILV - irá processar primeiro os segmentos cujos comprimentos são mais uniformes.

Nos grupos contendo blocos mais vazios (G1 a G3), processar primeiro os segmentos com a maior variância dos comprimentos - como MALV faz - terá como consequência que a maioria dos pixels tenderão a ser codificados de forma mais eficiente, uma vez que estão localizados nos segmentos mais cheios. Após o primeiro processamento 1D, a informação principal desses segmentos mais longos estará concentrada nos coeficientes de baixa frequência (os de alta frequência irão tender a zero). Ao contrário, os segmentos mais curtos serão codificados com menor eficiência, produzindo coeficientes de alta frequência com valores significativos. Antes do processamento na segunda direção em um bloco específico, os segmentos contendo os coeficientes na primeira direção de processamento serão alinhados com a borda do bloco, deslocando-se no sentido da direção preferencial (para cima, no caso da direção preferencial ser a vertical, ou para a esquerda, no caso da direção preferencial ser a horizontal). Os segmentos mais curtos na segunda direção de processamento, que serão codificados com menor eficiência, irão conter os coeficientes de alta frequência resultantes do processamento dos

segmentos mais cheios na primeira direção. Como esses coeficientes tendem a zero, serão menos significativos para os resultados finais de codificação.

Em contraposição à estratégia MALV, se MILV for usada nos grupos G1 a G3, a eficiência de codificação tende a ser baixa, porque os segmentos na primeira direção de processamento apresentarão comprimentos mais uniformes. Isso significa que todos eles serão pouco preenchidos por pixels do objeto (não haverá concentração de pixels do objeto em segmentos longos), resultando em coeficientes de alta frequência com valores significativos.

Entretanto, essa análise é extremamente dependente da matriz de alocação de bits gerada para cada objeto específico. Se o objeto apresenta baixas texturas em todos os grupos, como *PEPPER*, as variâncias de praticamente todos os coeficientes será baixa e a alocação de bits será mais ou menos uniforme entre os coeficientes de baixa e alta frequência, principalmente a taxas médias e altas. Nesse caso, a estratégia MILV poderá se tornar preferível à estratégia MALV nos grupos de blocos pouco ou medianamente preenchidos, ao processar primeiro os segmentos com comprimentos mais uniformes. Esses segmentos, por apresentarem textura baixa, poderão ser codificados de forma eficiente, mesmo tratando-se de transformadas 1D de dimensão reduzida. Além disso, esses segmentos produzirão coeficientes que estarão mais correlatados aos coeficientes de mesmo índice dos segmentos vizinhos (pois serão resultado da transformação de segmentos de comprimentos semelhantes e de baixa textura), o que aumentará a eficiência das transformadas 1D na segunda direção de processamento.

Para melhor visualização das análises a serem apresentadas, a Figura 6.1 mostra mais uma vez o indicador TNPO para os objetos descritos no Capítulo 4.

A Figura 6.2 mostra o desempenho de codificação dos blocos do grupo G1 de *CHILDREN*, que é o objeto com a maior textura nesse grupo. As Figuras 6.3 e 6.4 apresentam os resultados para os blocos do grupo G3 para objetos de baixa (*AKYIO*) e alta (*WEATHER*) textura, respectivamente. Nota-se que em todas as três situações, MALV é a estratégia mais eficiente, chegando a ganhos de RPR de cerca de 15 dB em relação a MILV.

A análise do desempenho de codificação dos grupos G1 a G3 de *PEPPER* usando MILV e MALV deve ser mais cuidadosa, como mencionado anteriormente. Tratando-se de um objeto de textura muito suave em todos os grupos, a matriz de alocação de bits será razoavelmente uniforme e MILV tenderá a ser mais eficiente que MALV. Em G1, onde os blocos são praticamente vazios, a diferença entre MILV e MALV não é perceptível (Figura 6.5). Em G2, já é possível notar a superioridade de MILV (Figura 6.6)

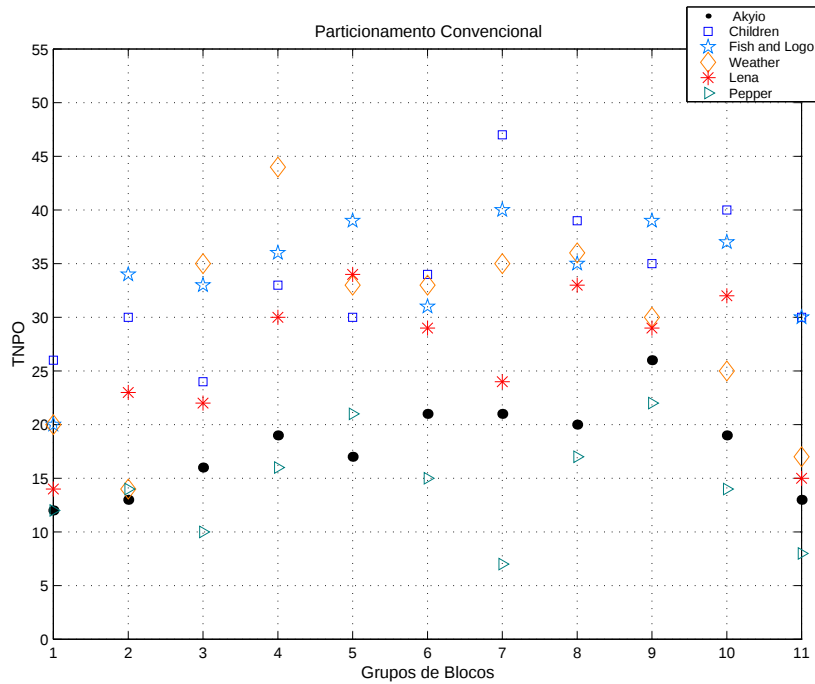


Figura 6.1: Textura de acordo com o Número de Pixels do Objeto - TNPO

e somente em G3, para o qual o indicador TNPO apresenta um valor extremamente baixo, MALV torna-se preferível a MILV (Figura 6.7).

Os grupos G4 a G10 estarão razoavelmente cheios e os segmentos do objeto tenderão a apresentar eficiência de codificação de média para alta. A estratégia MILV prioriza a direção onde os segmentos têm comprimentos mais uniformes, produzindo após o primeiro processamento 1D coeficientes das transformadas mais fortemente correlatados àqueles de mesmo índice nos segmentos vizinhos. Por outro lado, se MALV fosse usada nos grupos G4 a G10, a maioria dos pixels do objeto estariam localizados nos segmentos mais cheios, que tendem a produzir uma maior eficiência de codificação. Contudo, esse argumento é importante somente para blocos de baixa textura. Para texturas altas, produzir coeficientes de mesmo índice mais correlatados será mais importante que alocar a maior parte dos pixels do objeto nos segmentos mais cheios. Conseqüentemente, de G4 a G10, MALV tende a proporcionar resultados melhores para segmentos de baixa textura, enquanto MILV é superior nos segmentos de alta textura.

Ainda no caso *PEPPER*, são mostradas nas Figuras 6.8 e 6.9 as superioridades de MALV e MILV, respectivamente. Note-se que a textura de G5, apesar de não ter um valor alto, assume um dos maiores valores dentre os grupos de *PEPPER*, o que torna a estratégia MILV mais eficiente.

As Figuras 6.10 a 6.15 mostram os desempenhos do grupo G7 para todos os seis objetos. Para *PEPPER* (Figura 6.10) e *AKYIO* (Figura 6.11), que

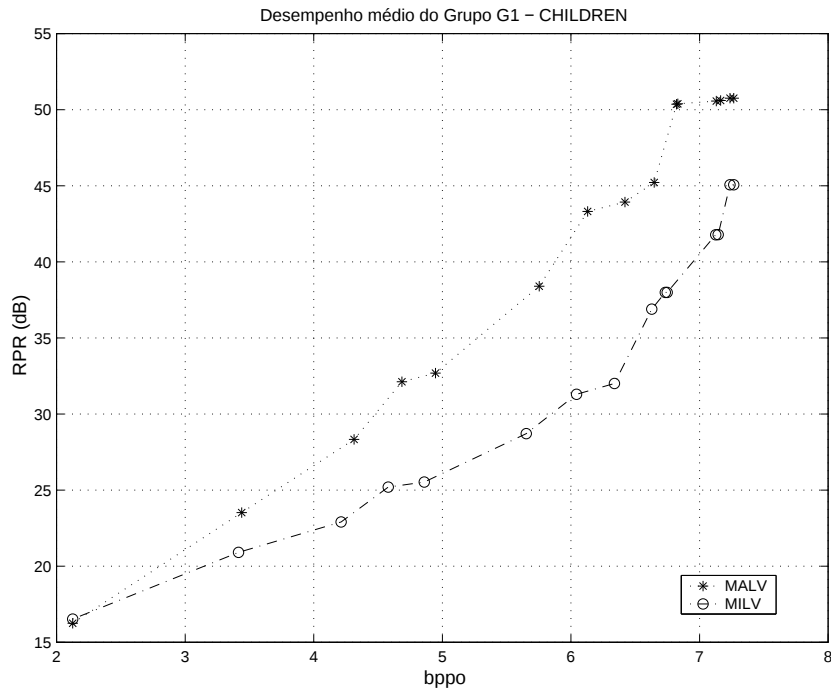


Figura 6.2: Desempenho médio dos blocos do grupo G1 em *CHILDREN*

têm texturas baixas, MALV supera MILV. Para *CHILDREN* (Figura 6.12), devido à textura extremamente alta, MILV é muito melhor que MALV. Para *WEATHER* (Figura 6.13), *FISH AND LOGO* (Figura 6.14) e *LENA* (Figura 6.15), que apresentam texturas médias, pode ser percebido que os desempenhos dos algoritmos MALV e MILV, em geral, se alternam, sendo mais difícil determinar a adequação de um ou outro.

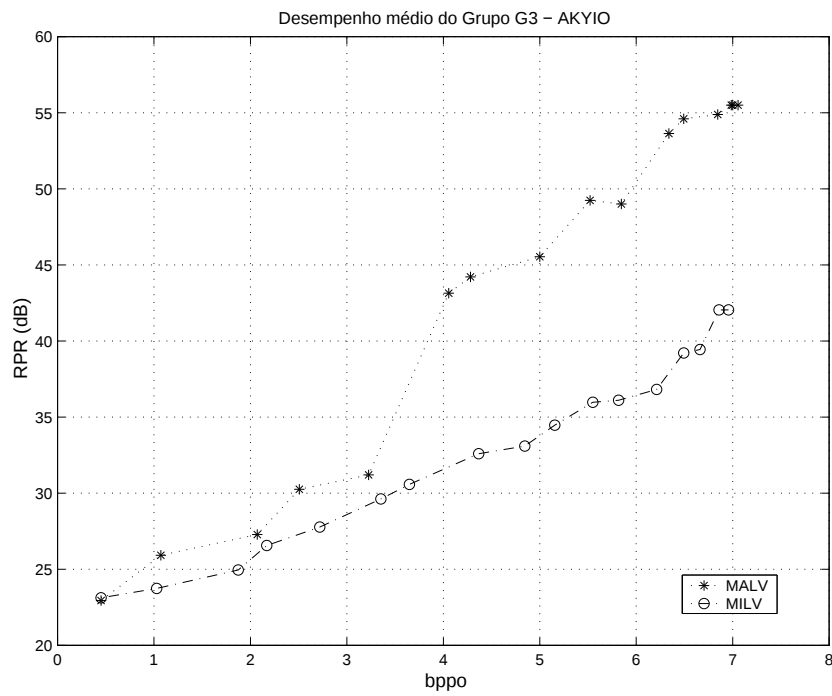


Figura 6.3: Desempenho médio dos blocos do grupo G3 em *AKYIO*

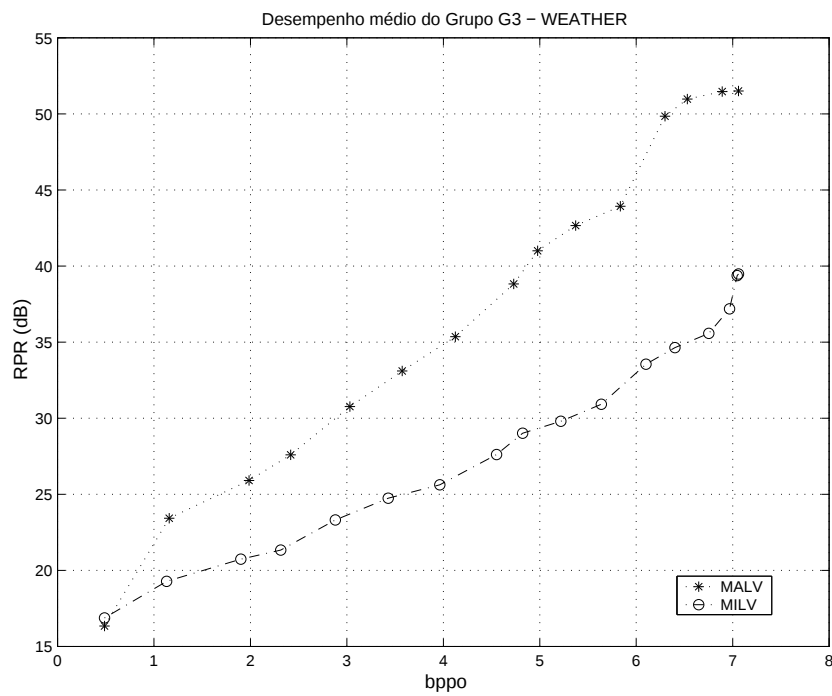


Figura 6.4: Desempenho médio dos blocos do grupo G3 em *WEATHER*

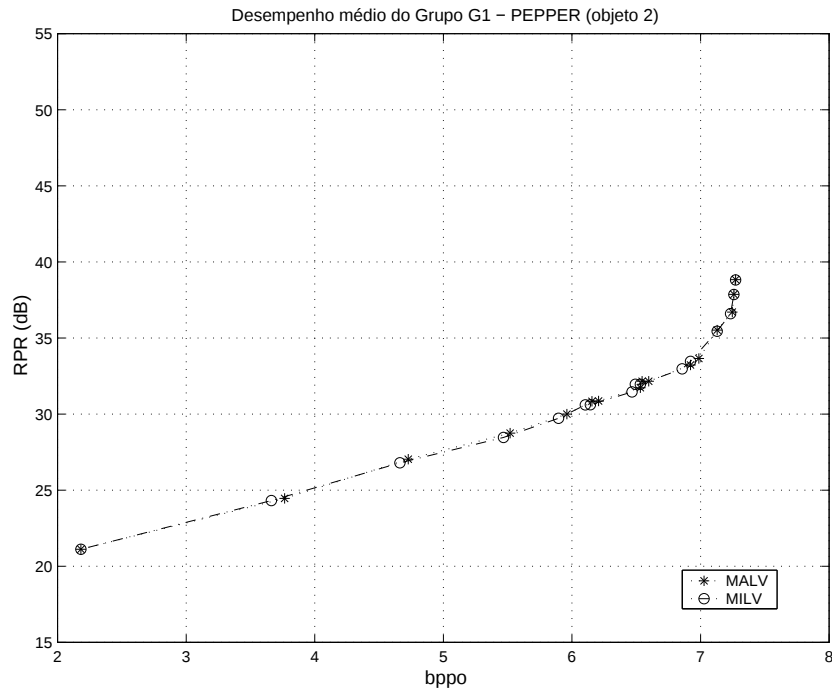


Figura 6.5: Desempenho médio dos blocos do grupo G1 em *PEPPER*

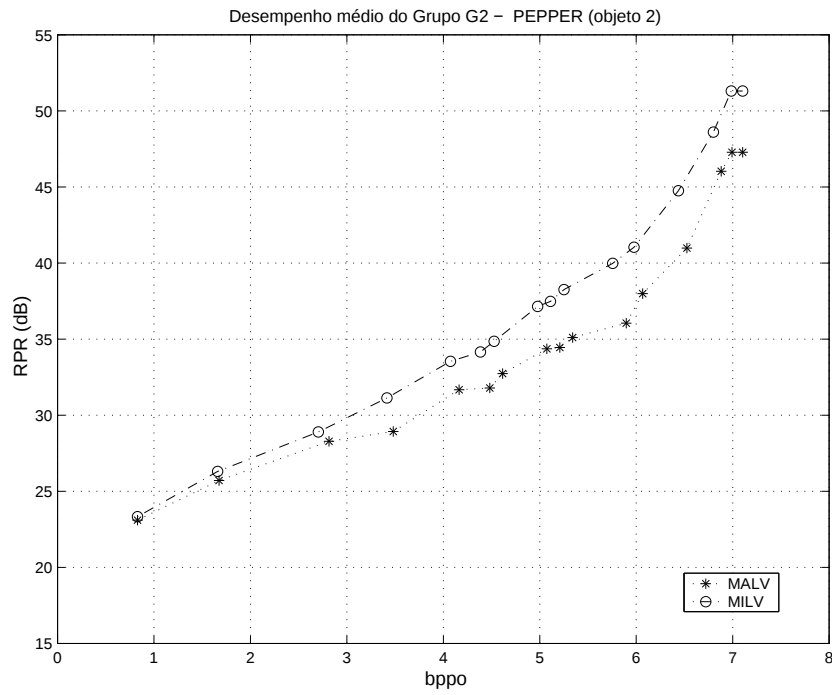


Figura 6.6: Desempenho médio dos blocos do grupo G2 em *PEPPER*

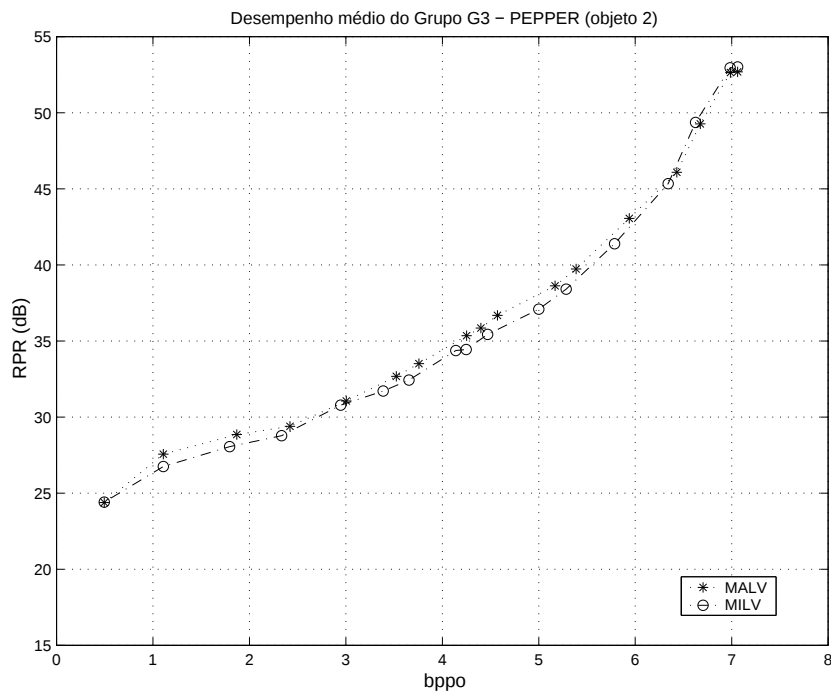


Figura 6.7: Desempenho médio dos blocos do grupo G3 em *PEPPER*

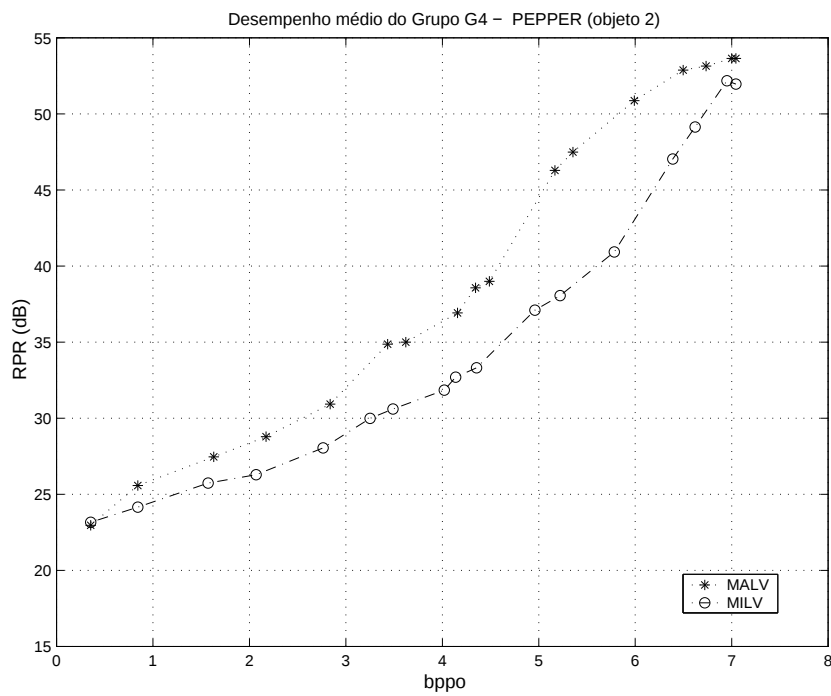


Figura 6.8: Desempenho médio dos blocos do grupo G4 em *PEPPER*

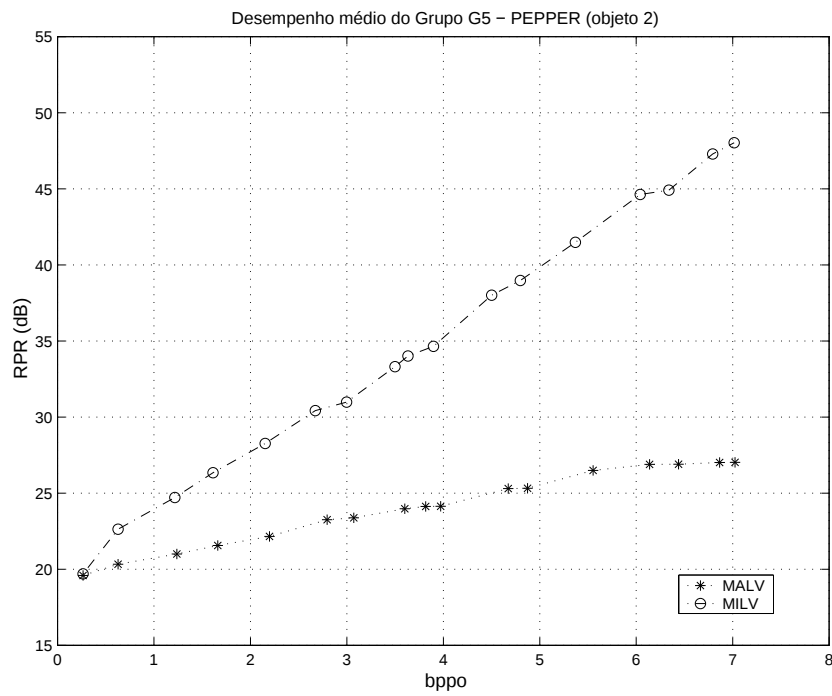


Figura 6.9: Desempenho médio dos blocos do grupo G5 em *PEPPER*

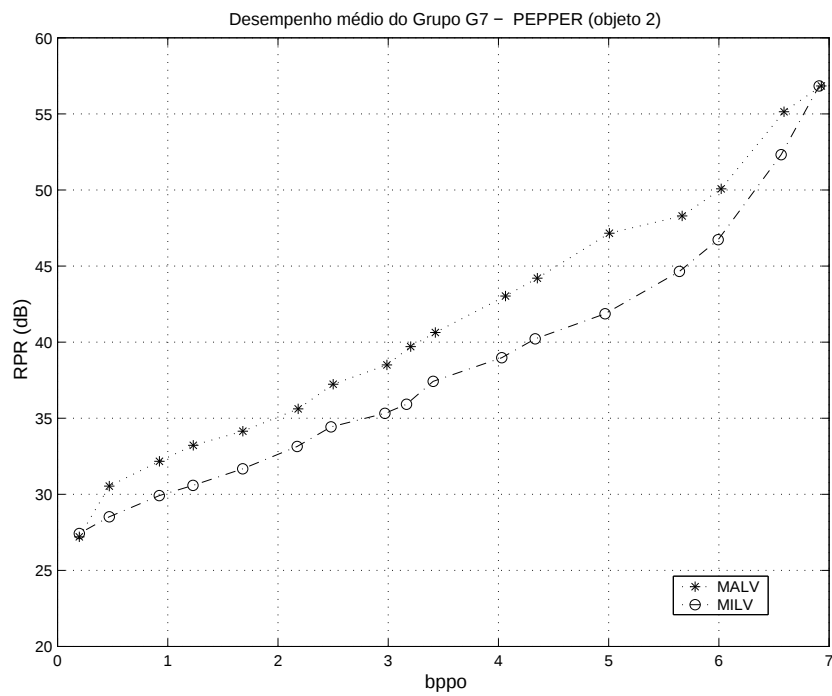


Figura 6.10: Desempenho médio dos blocos do grupo G7 em *PEPPER*

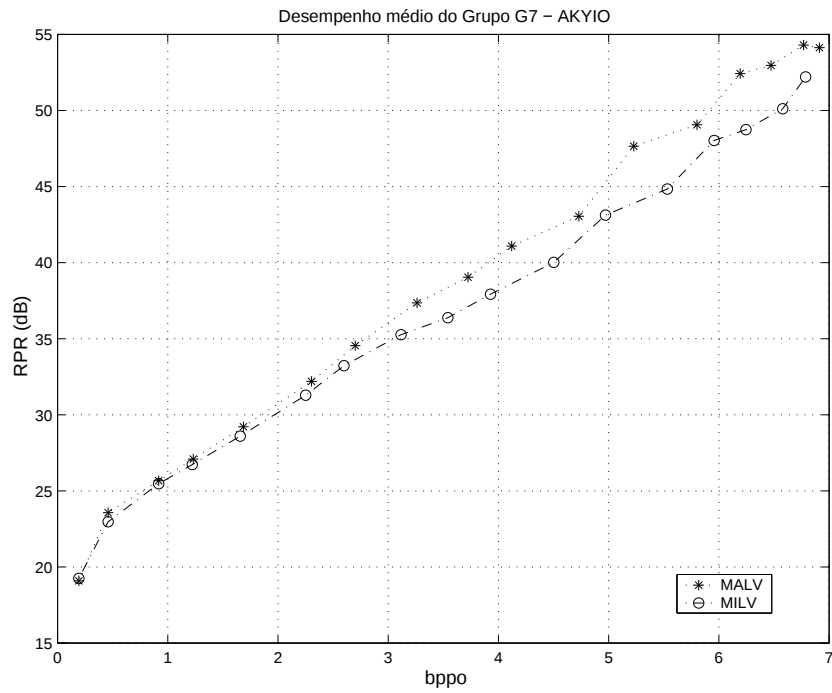


Figura 6.11: Desempenho médio dos blocos do grupo G7 em *AKYIO*

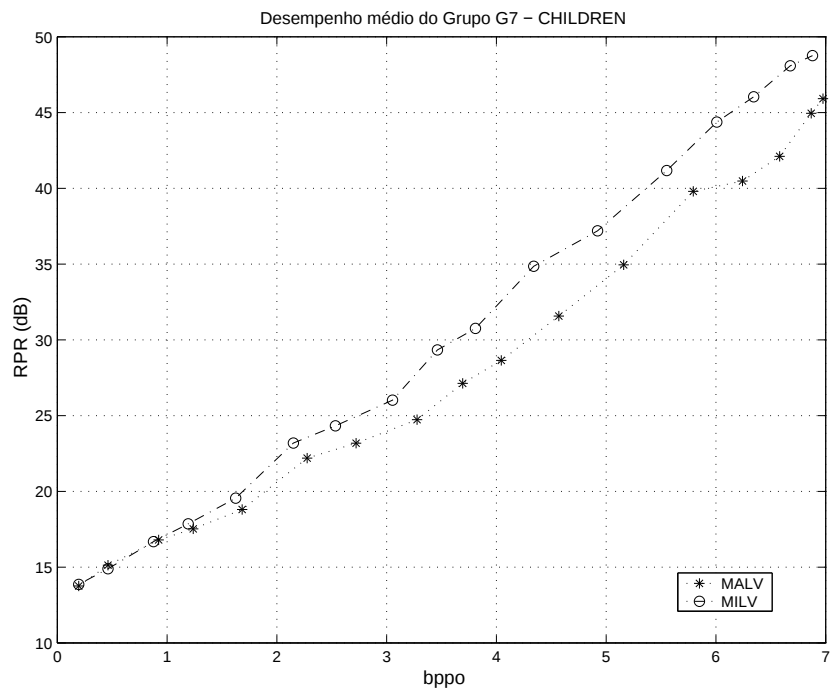


Figura 6.12: Desempenho médio dos blocos do grupo G7 em *CHILDREN*

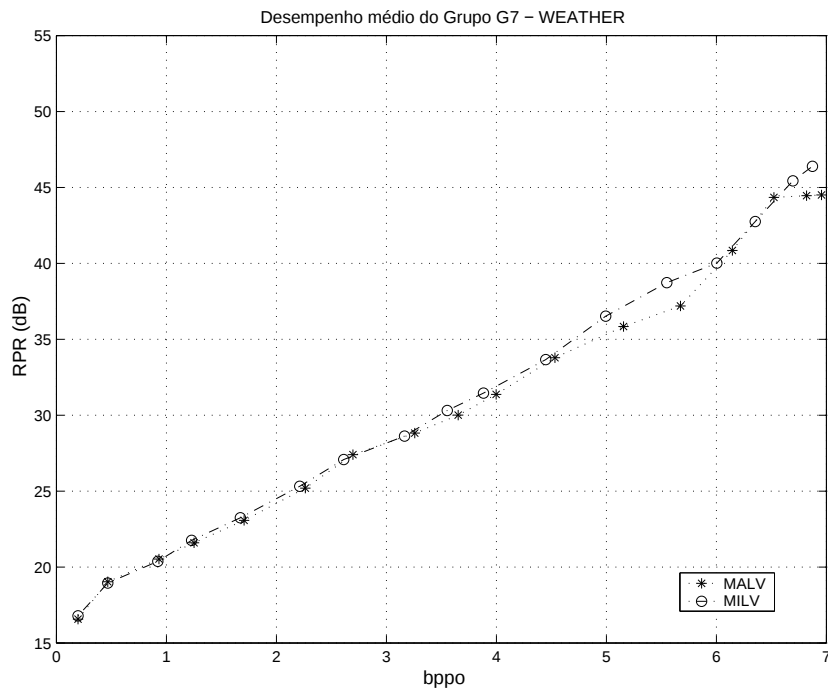


Figura 6.13: Desempenho médio dos blocos do grupo G7 em *WEATHER*

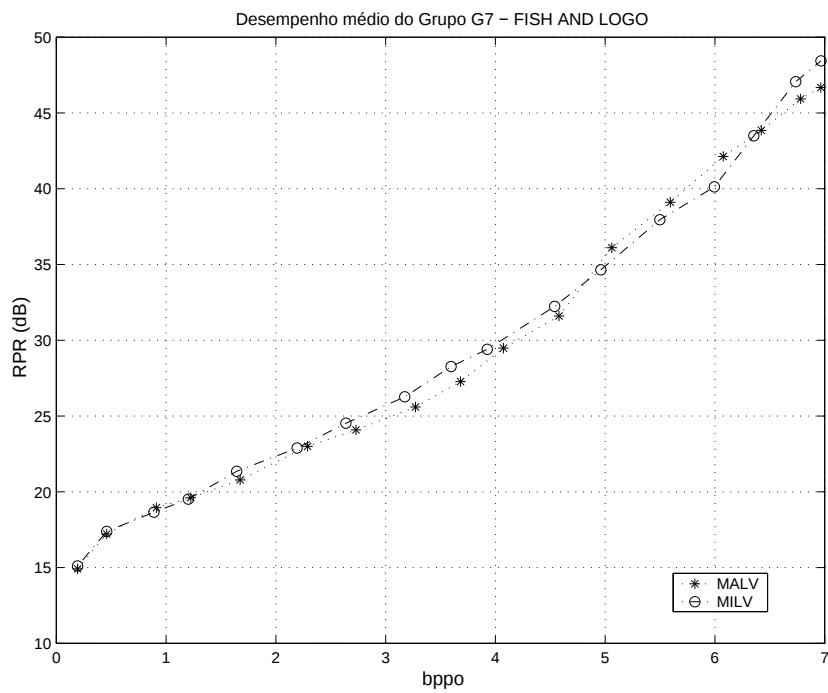


Figura 6.14: Desempenho médio dos blocos do grupo G7 em *FISH AND LOGO*

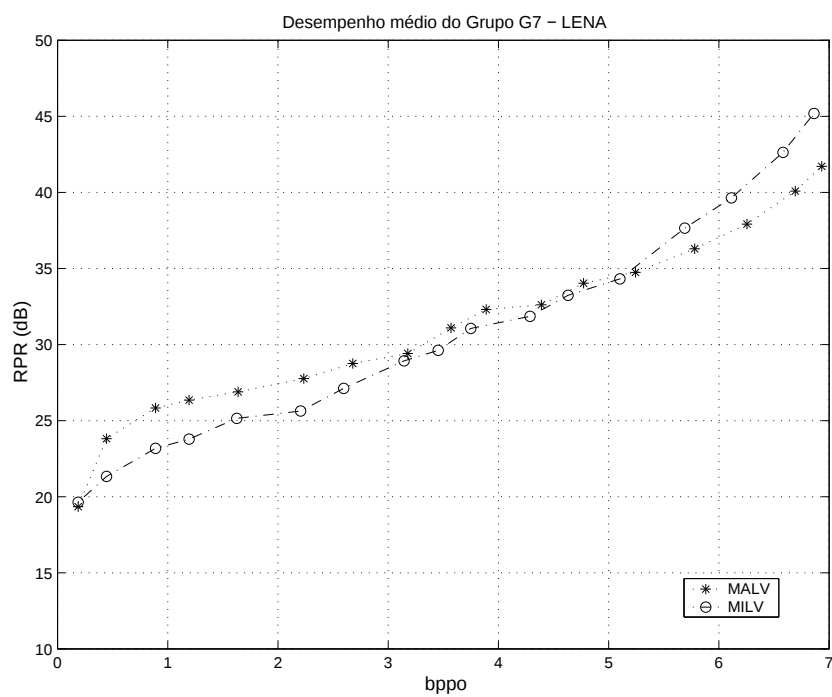


Figura 6.15: Desempenho médio dos blocos do grupo G7 em *LENA*

Quando os blocos estão mais preenchidos com pixels do objeto, a diferença entre MALV and MILV tende a ser menos significativa e os valores de textura acima dos quais MILV torna-se melhor que MALV normalmente é mais alto. Entretanto, a análise apresentada anteriormente continua valendo. Para ilustrar, são mostradas as curvas de desempenho do grupo G9 nas Figuras 6.16 a 6.19. Nesse grupo, para os objetos com textura mais alta (como *WEATHER*, na Figura 6.16), MILV tende a ser o melhor esquema. Já para *PEPPER* (Figura 6.17), *AKYIO* (Figura 6.18) e *LENA* (Figura 6.19), que são os objetos de menor textura no grupo, MALV supera MILV.

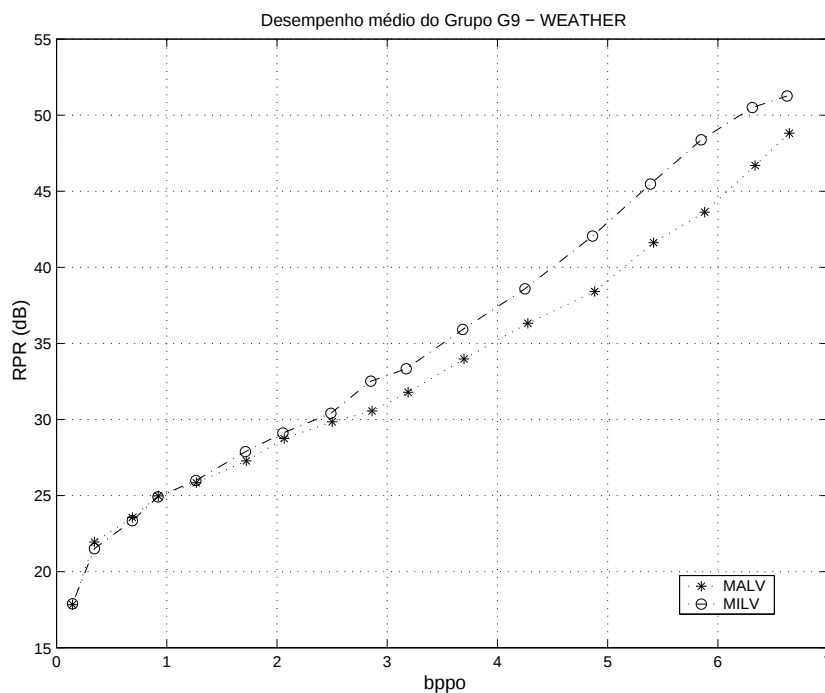


Figura 6.16: Desempenho médio dos blocos do grupo G9 em *WEATHER*

A partir desses resultados e análises preliminares, será proposto na seção seguinte um esquema híbrido que permita selecionar adaptativamente a melhor estratégia para cada grupo de blocos de contorno, tomando por base o indicador morfológico TNPO.

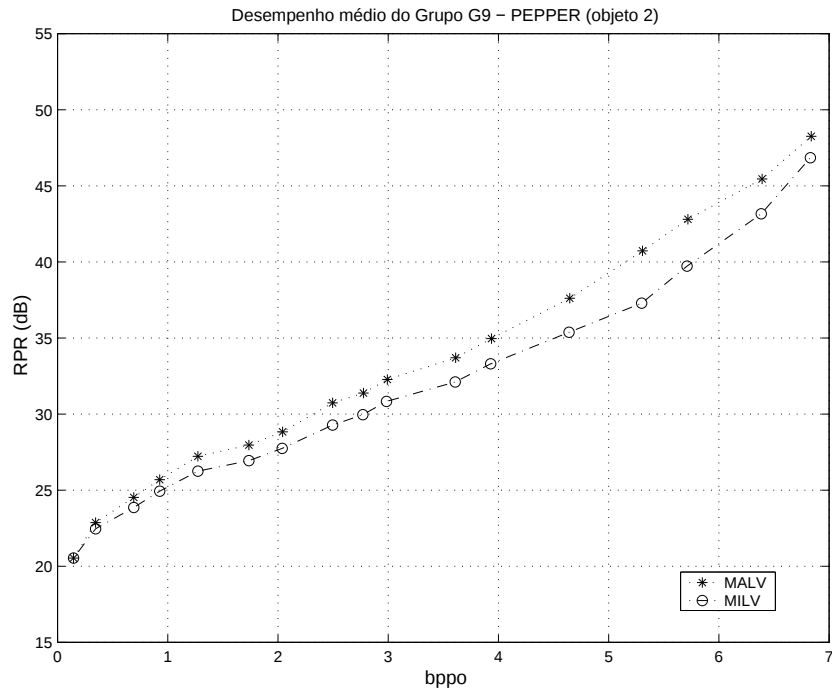


Figura 6.17: Desempenho médio dos blocos do grupo G9 em *PEPPER*

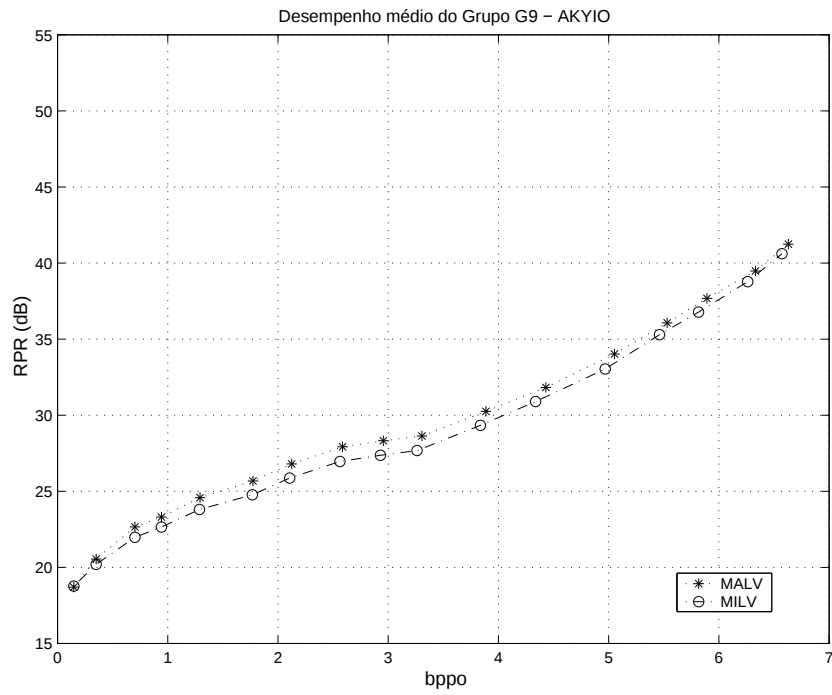


Figura 6.18: Desempenho médio dos blocos do grupo G9 em *AKYIO*

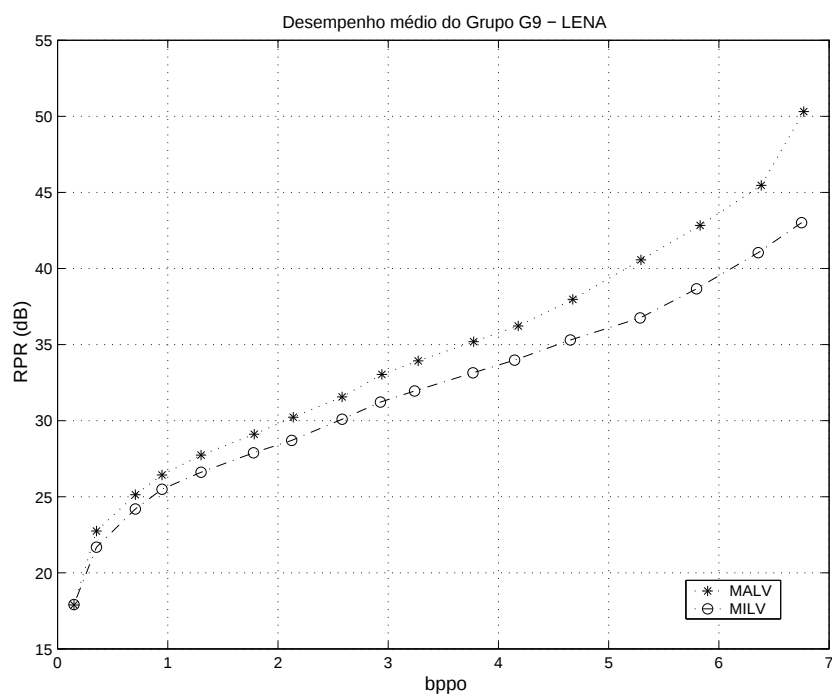


Figura 6.19: Desempenho médio dos blocos do grupo G9 em *LENA*

6.3

Algoritmo Híbrido para a SA-DCT: Proposição e Resultados

O objetivo do esquema híbrido proposto nesta seção é realizar a escolha adaptativa da direção preferencial para o processamento 1D da SA-DCT, de acordo com o indicador morfológico TNPO. Sempre que houver uma boa correlação entre os desempenhos de MALV e MILV e o indicador TNPO para um objeto específico, será dada prioridade às estratégias que utilizem apenas a informação sobre as variâncias dos comprimentos dos segmentos do objeto - MILV e MALV. Nas situações em que a determinação dos limiares de TNPO para o chaveamento entre essas estratégias é mais difícil, será usada a estratégia MACES (descrita na Seção 5.3).

Para o esquema EI-DCT híbrido (Tabela 5.1), a estratégia MACES pode eventualmente ser escolhida em qualquer dos grupos de blocos de contorno de $G1$ a $G10$. Contudo, no caso da SA-DCT, como o número de bits alocados para os blocos pouco preenchidos é pequeno se comparado aos blocos mais preenchidos, o uso da estratégia MACES não é adequado, pois a transmissão de 1 bit adicional por bloco de contorno representa uma sobre-taxa significativa. Por esse motivo, a estratégia MACES só será utilizada no esquema híbrido para a SA-DCT nos blocos medianamente preenchidos. Nos blocos mais cheios, de $G8$ a $G10$, a eficiência de codificação da SA-DCT é alta e o ganho proporcionado pela estratégia MACES na determinação da direção preferencial nos grupos de blocos com textura média não compensa a necessidade de transmitir 1 bit adicional por bloco. Para esses grupos de blocos cheios, normalmente a estratégia MALV, que concentra a maioria dos pixels em segmentos longos, tenderá a ser mais eficiente que as estratégias MILV e MACES (vide Figura 6.20), a não ser que a textura seja muito elevada, casos em que MILV é mais indicada (vide Figura 6.21).

O modo de chaveamento do algoritmo híbrido baseou-se nas observações e justificativas apresentadas na Seção 6.2, que apontaram a estratégia MALV como a mais eficiente para os grupos de blocos mais vazios, de $G1$ a $G3$, ao passo que nos blocos medianamente preenchidos, de $G4$ a $G7$, a indicação do uso de MILV ou MALV está fortemente condicionada à textura. Nesse caso, para grupos de blocos com textura baixa, a estratégia MALV é preferível à MILV, mas para grupos com textura alta, a estratégia MILV é mais adequada. Para grupos com textura média, foi mostrado na Seção 6.2 que os desempenhos de MALV e MILV se alternam, sendo mais difícil a decisão por uma ou pela outra. Por esse motivo, nos grupos de textura média, a estratégia MACES proporciona ganhos de codificação su-

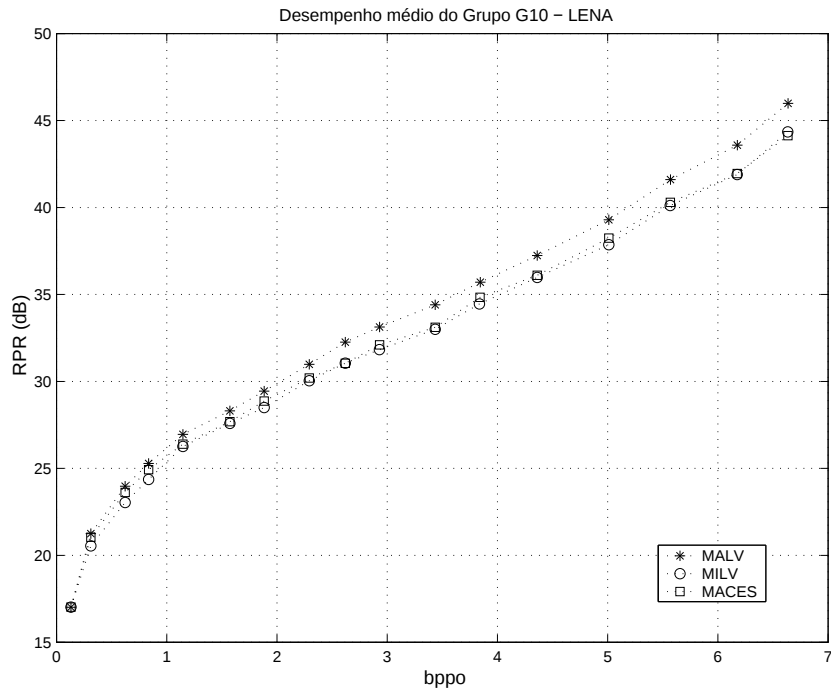


Figura 6.20: Desempenho médio dos blocos do grupo G10 em *LENA*

periores a MILV e MALV. Como exemplo, a Figura 6.22 mostra os ganhos de codificação de MALV, MILV e MACES para o grupo *G7* de *AKYIO*, que apresenta textura média.

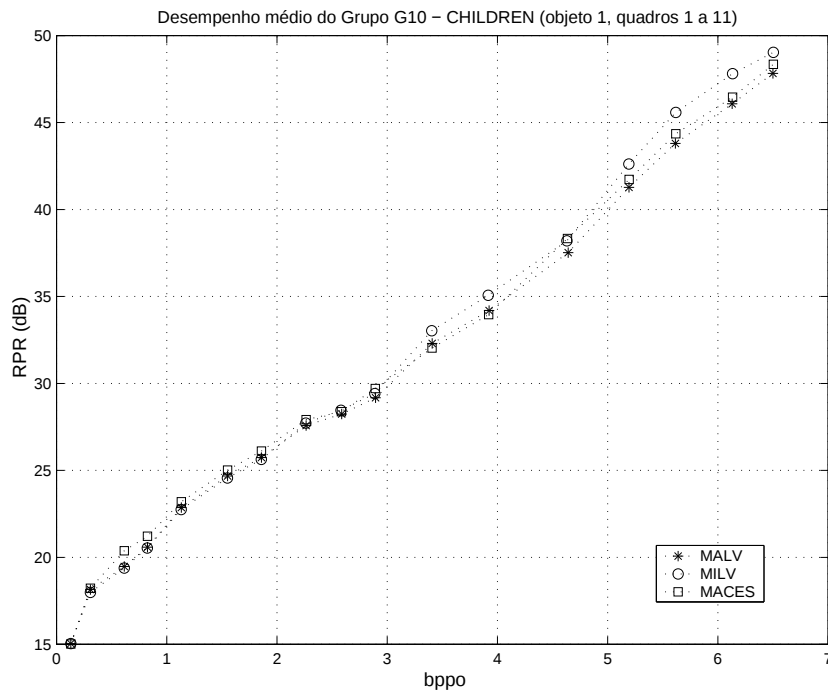


Figura 6.21: Desempenho médio dos blocos do grupo G10 em *CHILDREN* (objeto 1, quadros 1 a 11)

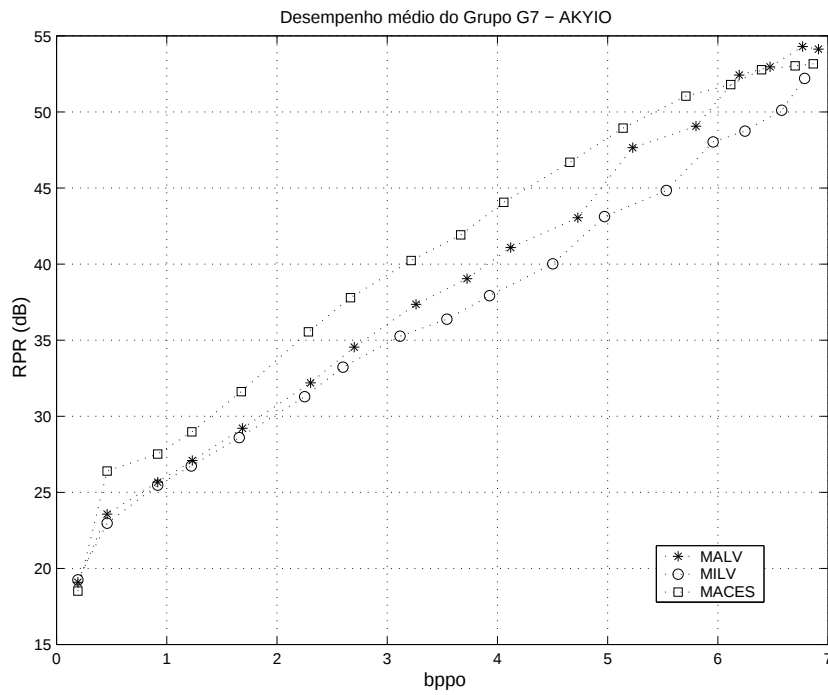


Figura 6.22: Desempenho médio dos blocos do grupo G7 em *AKYIO*

Uma síntese das principais considerações para a escolha da direção preferencial da SA-DCT é apresentada a seguir:

- Grupos de blocos mais vazios (G1 a G3):
 - Com MALV: a maioria dos pixels será codificada de forma mais eficiente após o 1º. processamento 1D, concentrando a informação principal nos coeficientes de baixa frequência;
 - Com MILV: a codificação resultante do 1º. processamento 1D será menos eficiente (não haverá concentração de pixels em segmentos longos), podendo resultar em coeficientes de alta frequência com valores significativos;
 - Pelas razões apontadas, MALV tende a ser mais eficiente. Contudo, a eficiência de MILV e MALV nos grupos de blocos pouco ou medianamente preenchidos também depende da alocação de bits e da correlação entre coeficientes de mesmo índice após o 1º. processamento 1D.
 - Em objetos com muito baixas texturas em todos os grupos, as variâncias de todos os coeficientes serão baixas e portanto a distribuição dos bits entre coeficientes de alta e baixa frequência será mais uniforme (principalmente a taxas médias e altas). Portanto, MILV pode se tornar preferível a MALV nos blocos pouco ou medianamente preenchidos desses objetos, uma vez que produz coeficientes mais correlatados aos coeficientes de mesmo índice em segmentos vizinhos após o 1º. processamento.
- À medida em que os blocos tornam-se mais cheios (G4 a G10), a comparação $MALV \times MILV$ deve levar em consideração que:
 - Com MALV: a maioria dos pixels estará concentrada em segmentos mais cheios (argumento importante para grupos de blocos de textura baixa);
 - Com MILV: após o 1º. processamento, ocorrerá a produção de coeficientes mais correlatados aos coeficientes de mesmo índice de segmentos vizinhos (argumento importante para grupos de blocos de textura alta);
 - Em casos de grupos de blocos medianamente preenchidos com texturas na faixa intermediária, as eficiências de MILV e MALV irão alternar-se, sendo mais difícil relacioná-las ao indicador TNPO.

- Comparando-se os desempenhos de MILV, MALV e MACES em todos os grupos de blocos, observamos ainda que:
 - MACES não é adequada para blocos vazios (uma vez que constitui uma sobre-taxa expressiva), sendo utilizada apenas nos blocos medianamente preenchidos;
 - Nos blocos medianamente preenchidos, a indicação de MALV ou MILV está fortemente relacionada à textura (TNPO);
 - Nos blocos cheios, a eficiência de codificação é alta e MALV ou MILV devem ser usados preferencialmente a MACES, que requer a transmissão de 1 bit adicional por grupo.

Baseando-se nessas considerações, a Tabela 6.1 detalha a operação do esquema híbrido para a SA-DCT.

Tabela 6.1: O Algoritmo Híbrido para a SA-DCT

$g = G1$ a $G3$	$g = G4$ a $G7$	$g = G8$ e $G9$	$g = G10$
MALV	MILV, se $TNPO(g) \geq 30$ MACES, se $30 > TNPO(g) \geq 21$ MALV, se $TNPO(g) < 21$	MILV, se $TNPO(g) \geq 30$ MALV, se $TNPO(g) < 30$	MILV, se $TNPO(g) \geq 35$ MALV, se $TNPO(g) < 35$

Os resultados de codificação dos seis objetos de teste, usando o esquema adaptativo proposto, são apresentados nas Figuras 6.23 a 6.28 em termos das RPR dos blocos de contorno. Cada uma dessas figuras mostra os desempenhos de codificação das estratégias MALV, MILV, ‘COLUMNS-FIRST’ e o método HÍBRIDO proposto.

Como todas as estratégias para a determinação da direção preferencial de processamento da SA-DCT operam sobre o mesmo conjunto de pixels (ao contrário dos esquemas de DCT baseada em blocos, que geram blocos com os pixels externos ao objeto calculados de maneira diferente, antes da aplicação da DCT-2D), foi determinada apenas uma estatística dos coeficientes, a ser usada para a determinação de matrizes de alocação de bits únicas para todas as estratégias. Somente os pixels pertencentes ao objeto foram usados para o cálculo das estatísticas. Cada bloco de contorno contribuiu duas vezes para o cálculo das médias e variâncias, tomando-se ora a direção vertical, ora a horizontal, para a aplicação do primeiro conjunto de transformadas 1D. Obviamente, os blocos cheios, nos quais se aplica apenas a DCT-2D, entraram apenas uma vez para o cômputo geral das estatísticas.

Nas Figuras 6.23 a 6.25, observa-se que em *PEPPER*, *CHILDREN* e *FISH AND LOGO*, a estratégia MILV foi melhor que a estratégia MALV. No caso de *PEPPER*, a estratégia ‘COLUMNS-FIRST’ chega até a superar MALV. Já em *AKYIO* e *WEATHER* (Figuras 6.26 e 6.27), a estratégia MILV foi mais eficiente que MALV. No objeto de *LENA* (Figura 6.28), MILV, MALV e ‘COLUMNS-FIRST’ apresentaram desempenhos equivalentes. Observa-se ainda que a eficiência da estratégia ‘COLUMNS-FIRST’, que foi originalmente empregada no algoritmo da SA-DCT [67], é extremamente variável. Ela pode ser comparável ou superior à melhor das estratégias entre MILV e MALV em alguns casos, mas inferior a ambas em outros. Finalmente, verifica-se que para todos os seis objetos, o esquema híbrido mostrou-se equivalente ou sensivelmente superior à melhor entre as estratégias MILV, MALV e ‘COLUMNS-FIRST’.

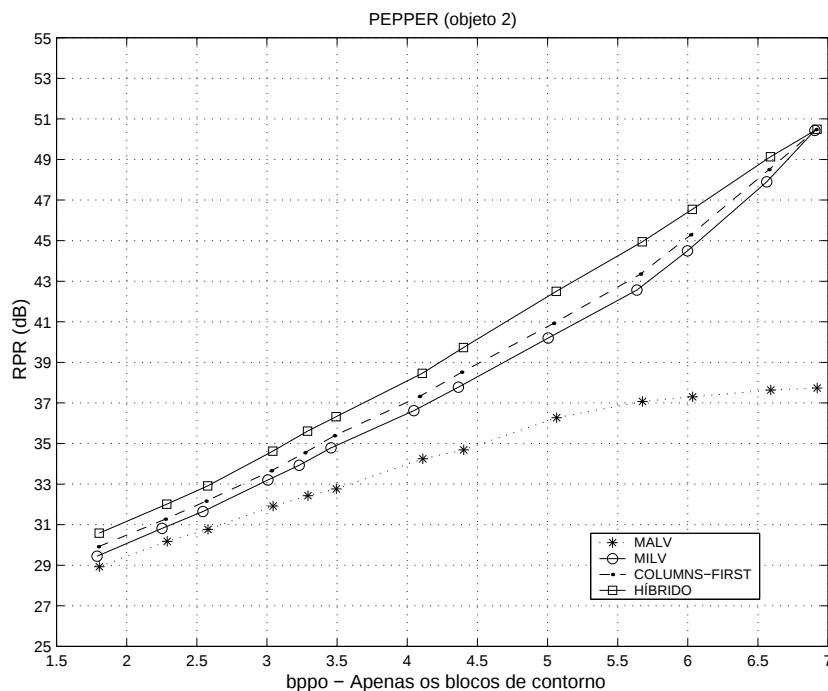


Figura 6.23: Resultados de codificação de *PEPPER* (objeto 2) com a SA-DCT

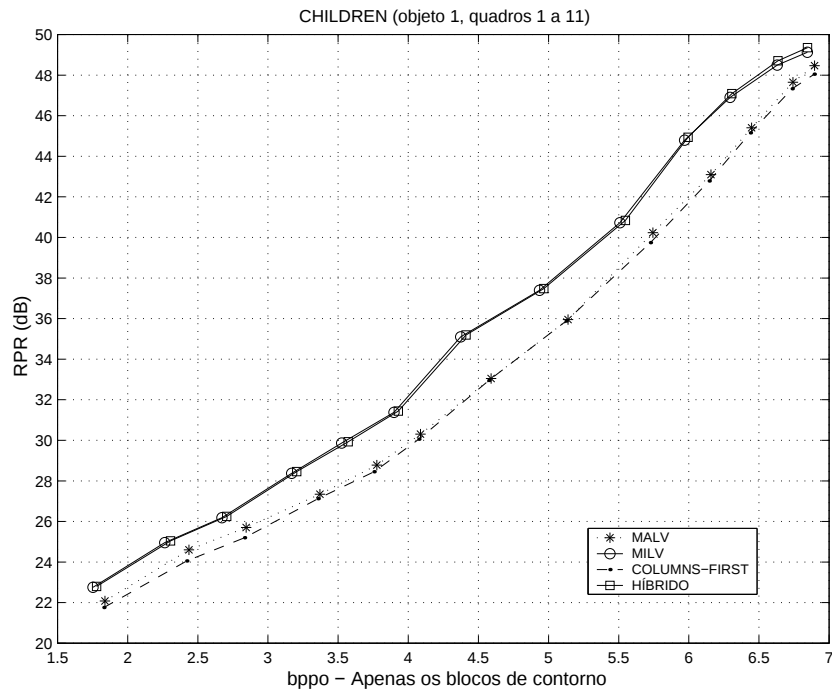


Figura 6.24: Resultados de codificação de *CHILDREN* (objeto 1, quadros 1 a 11) com a SA-DCT

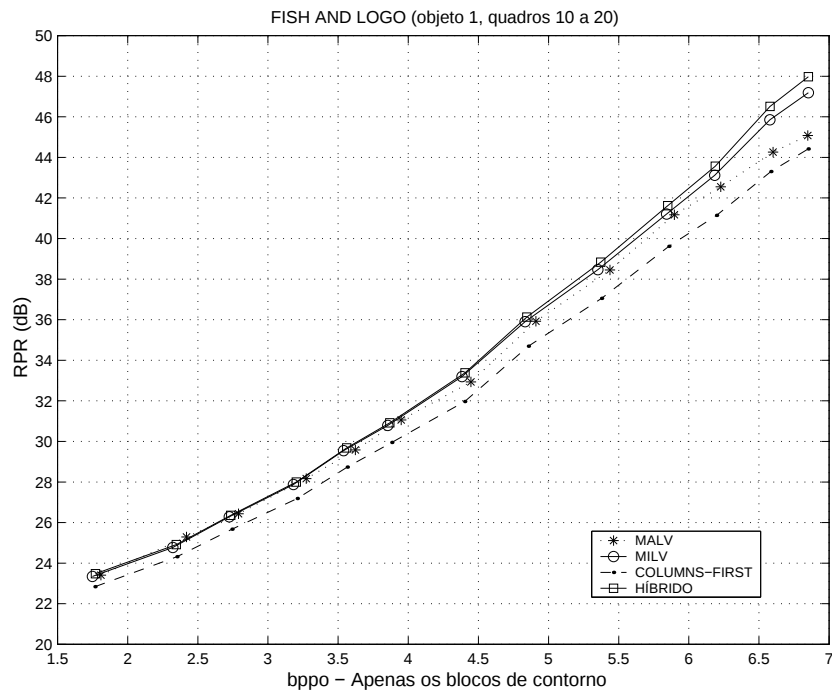


Figura 6.25: Resultados de codificação de *FISH AND LOGO* (objeto 1, quadros 10 a 20) com a SA-DCT

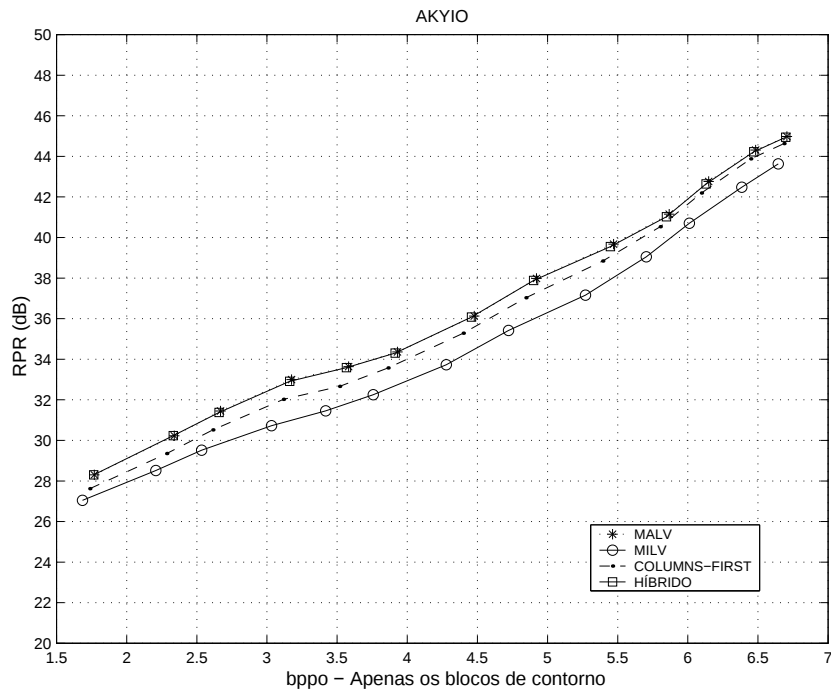


Figura 6.26: Resultados de codificação de *AKYIO* com a SA-DCT

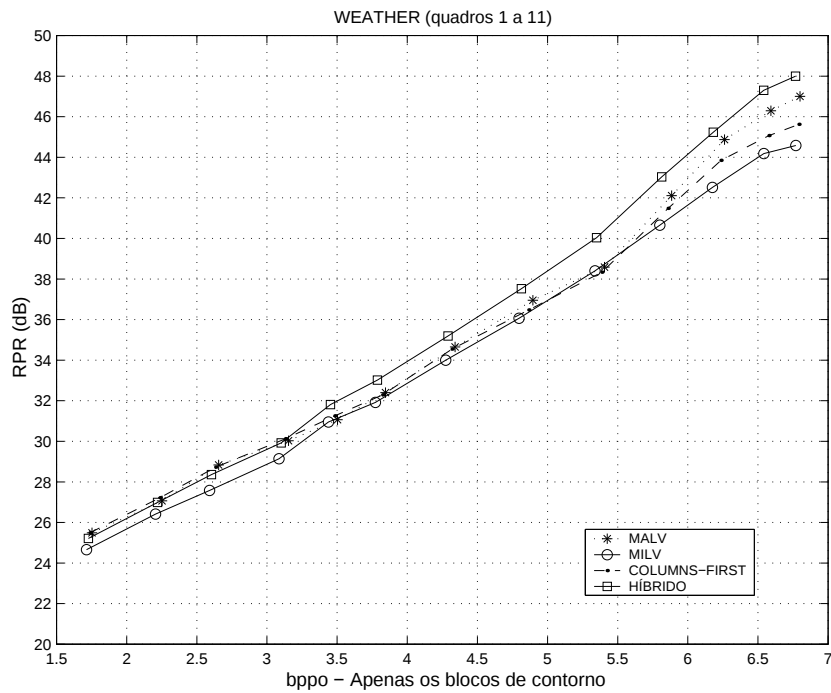


Figura 6.27: Resultados de codificação de *WEATHER* (quadros 1 a 11) com a SA-DCT

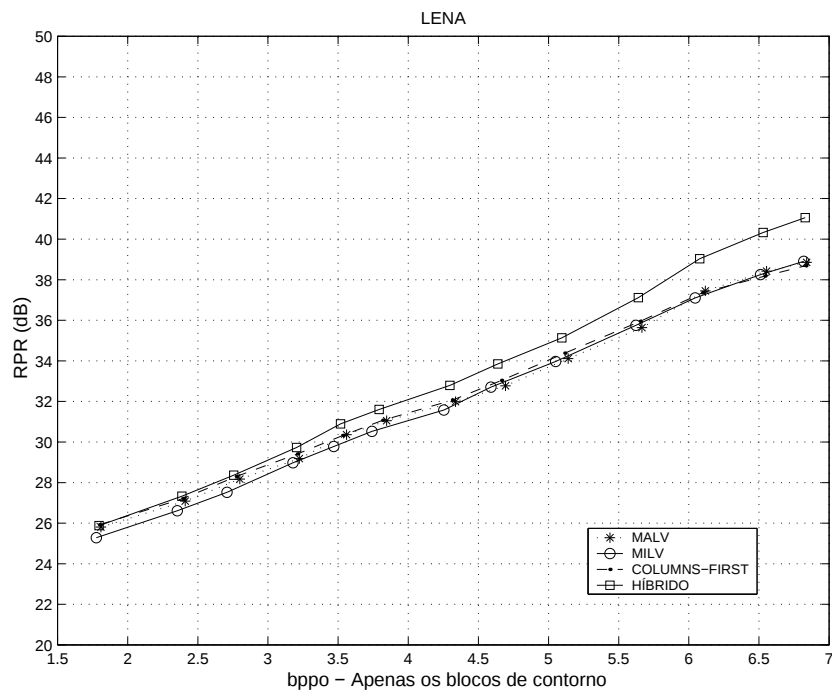


Figura 6.28: Resultados de codificação de *LENA* com a SA-DCT

6.4

O Emprego de Outros Particionamentos na SA-DCT

Na codificação por DCT baseada em blocos, observou-se na Seção 5.5 que a redução do número de blocos a codificar através do emprego dos esquemas de Particionamento Ortogonal Simplificado e Flexível Simplificado [94] proporcionaram um aumento do ganho de codificação, principalmente a taxas altas. Isso era esperado porque, reduzindo-se o número de blocos que contêm pixels do objeto, reduz-se proporcionalmente o número de pixels a serem codificados e, portanto, a taxa de *bppo* (número médio de bits por pixel do objeto) necessária para uma dada qualidade requerida. Isso foi visto no Capítulo 5 para a EI-DCT. Na codificação adaptativa à forma, contudo, o número de pixels a codificar é igual ao número de pixels do objeto e, conseqüentemente, a redução do número de blocos que contêm pixels do objeto não provoca nenhuma alteração na quantidade de pixels a serem codificados. As Figuras 6.29 a 6.34 mostram os ganhos de RPR (em dB) obtidos com as estratégias ‘HÍBRIDA’, ‘COLUMNS-FIRST’ e o critério que escolhe a direção que produz o menor produto $emq \times nbits$, aqui denominada estratégia ‘R-D pseudo-ÓTIMA’, ao empregarem os esquemas de particionamento Ortogonal Simplificado e Flexível Simplificado, em relação aos ganhos obtidos com a partição convencional. As Figuras 6.29 e 6.30 referem-se aos objetos de menor Área Efetiva, *CHILDREN* e *FISH AND LOGO*, ao passo que as demais referem-se a objetos com valores maiores desse parâmetro.

Apesar de em [76] a maior vantagem apontada para os esquemas de particionamento ter sido o aumento do ganho de codificação da SA-DCT através da concentração dos pixels do objeto em blocos mais cheios, observa-se pelas Figuras 6.29 a 6.34 que não existe a garantia de melhoria de eficiência de codificação para nenhuma das três estratégias. Essa afirmação pode ser feita inclusive para a estratégia ‘R-D pseudo-ÓTIMA’, que escolhe, por definição, a primeira direção de processamento que produza o menor produto taxa-distorção. Note-se que para essa estratégia, ambos os esquemas de particionamento, Ortogonal e Flexível, proporcionam ganhos variando entre -1 e 1 dB, na faixa de 0 a 7 *bppo*, sem a garantia de que esses esquemas sejam eficientes para uma determinada faixa de *bppo*. Basicamente, a eficiência de codificação em um bloco irá depender da textura do bloco, sendo maior para os blocos com textura menor. Ao empregar-se a estratégia ‘COLUMNS-FIRST’, também pode haver redução do ganho de codificação para algumas taxas de *bppo*, mesmo sabendo que todos os blo-

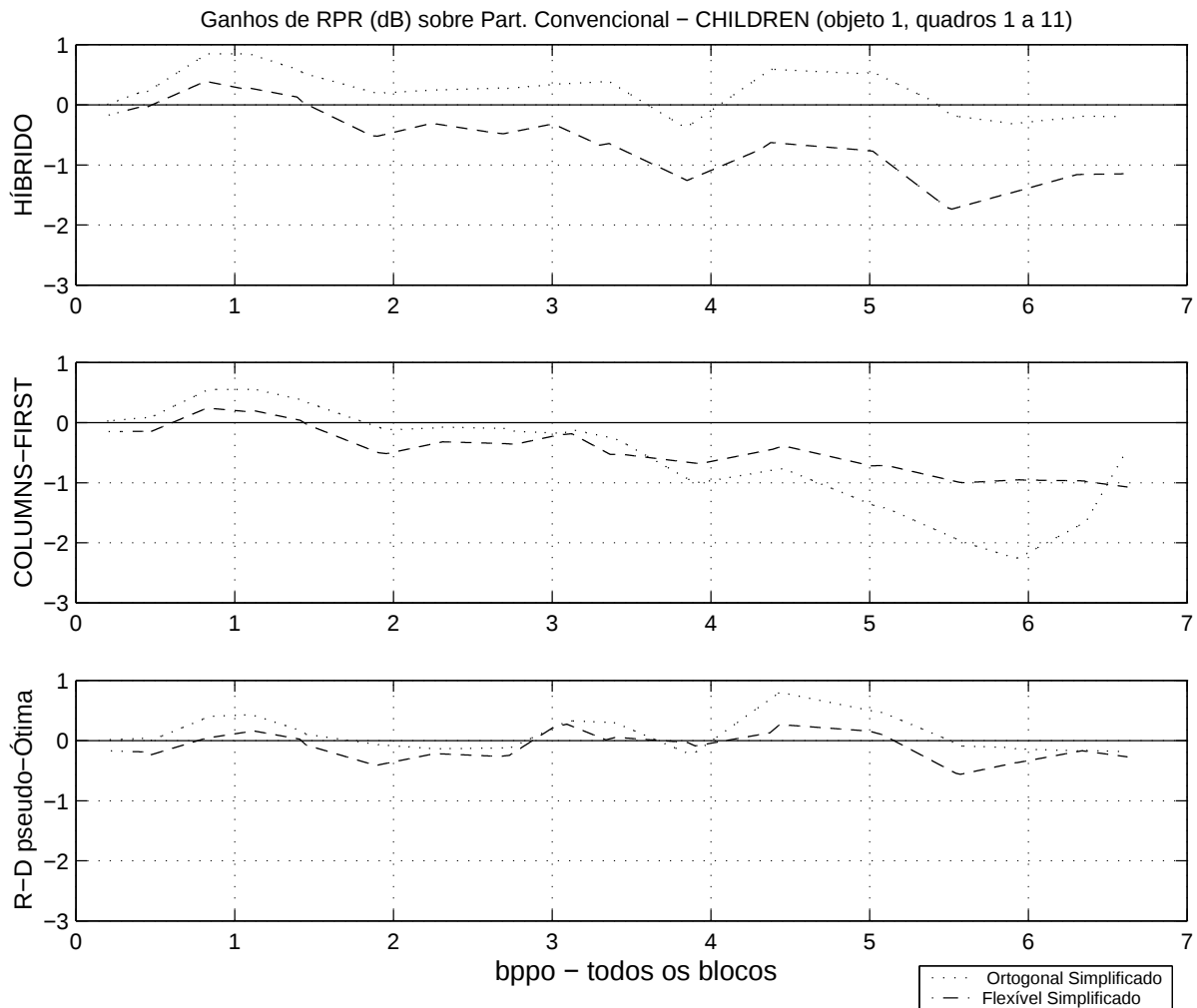


Figura 6.29: Estratégias ‘HÍBRIDA’, ‘COLUMNS-FIRST’ e ‘R-D pseudo-ÓTIMA’: ganhos de RPR (dB) sobre a partição convencional em *CHILDREN* (objeto 1, quadros 1 a 11)

cos escolherão como direção preferencial para o processamento dos blocos de contorno, independente do esquema de particionamento, a direção vertical (diferentemente da técnica ‘R-D pseudo-ÓTIMA’, que pode escolher quaisquer das direções como direção prioritária).

A incerteza quanto à melhoria do desempenho de codificação de ambas as estratégias ao empregarem os esquemas de particionamento alternativos - Ortogonal ou Flexível - deve-se à redistribuição dos pixels em um número de blocos menor, normalmente com texturas maiores que o conjunto de blocos na partição convencional, tendendo a reduzir o ganho médio de codificação. Essa redistribuição altera os indicadores morfológicos do objeto: tanto a Área Efetiva, quanto DNPO e TNPO. As variações da Área Efetiva segundo o tipo de particionamento foram mostradas na Seção 5.6. As

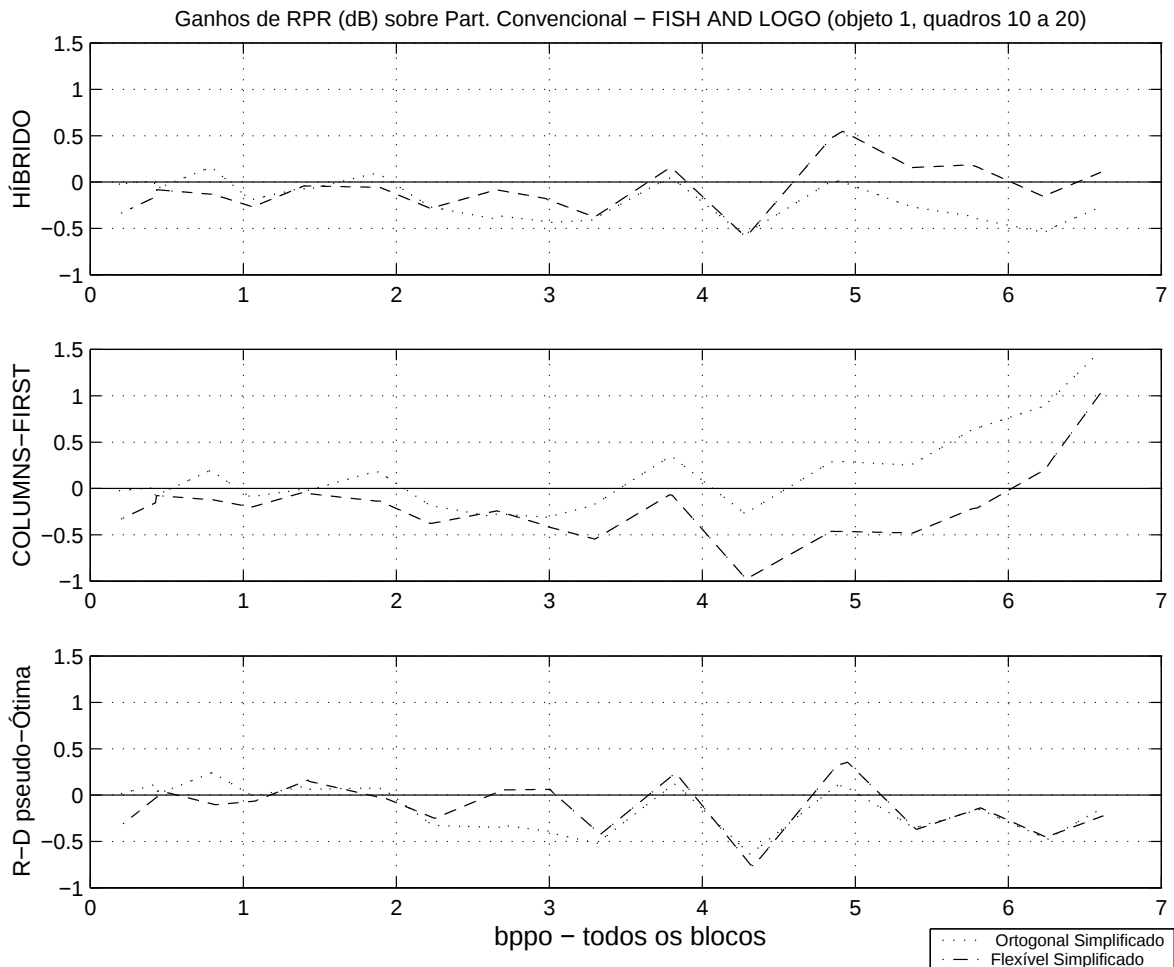


Figura 6.30: Estratégias ‘HÍBRIDA’, ‘COLUMNS-FIRST’ e ‘R-D pseudo-ÓTIMA’: ganhos de RPR (dB) sobre a partição convencional em *FISH AND LOGO* (objeto 1, quadros 10 a 20)

Figuras 6.35 a 6.37 mostram o indicador TNPO para os particionamentos convencional, Ortogonal Simplificado e Flexível Simplificado. Obviamente, também ocorrem variações com o indicador DNPO, que serão mostradas para casos específicos, posteriormente.

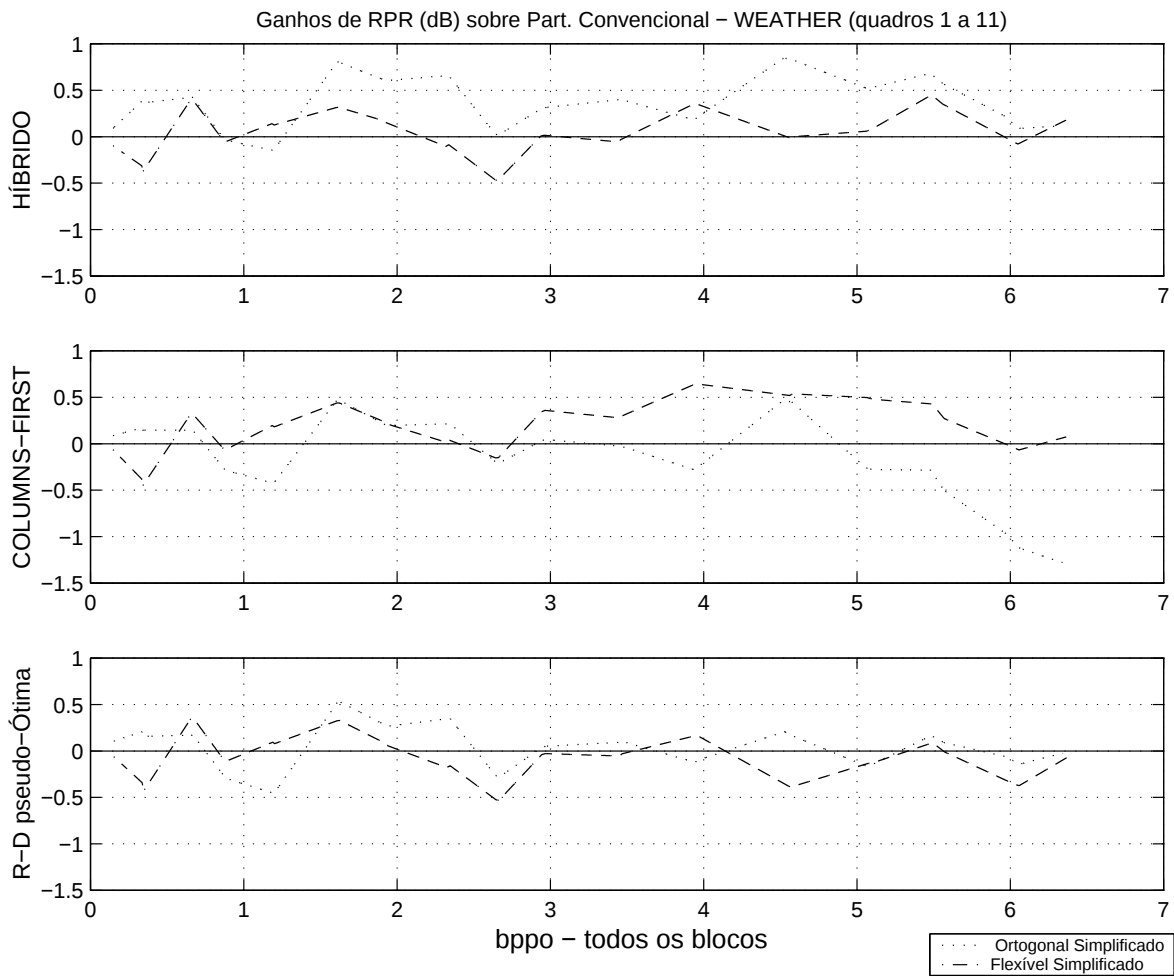


Figura 6.31: Estratégias ‘HÍBRIDA’, ‘COLUMNS-FIRST’ e ‘R-D pseudo-ÓTIMA’: ganhos de RPR (dB) sobre a partição convencional em *WEATHER* (quadros 1 a 11)

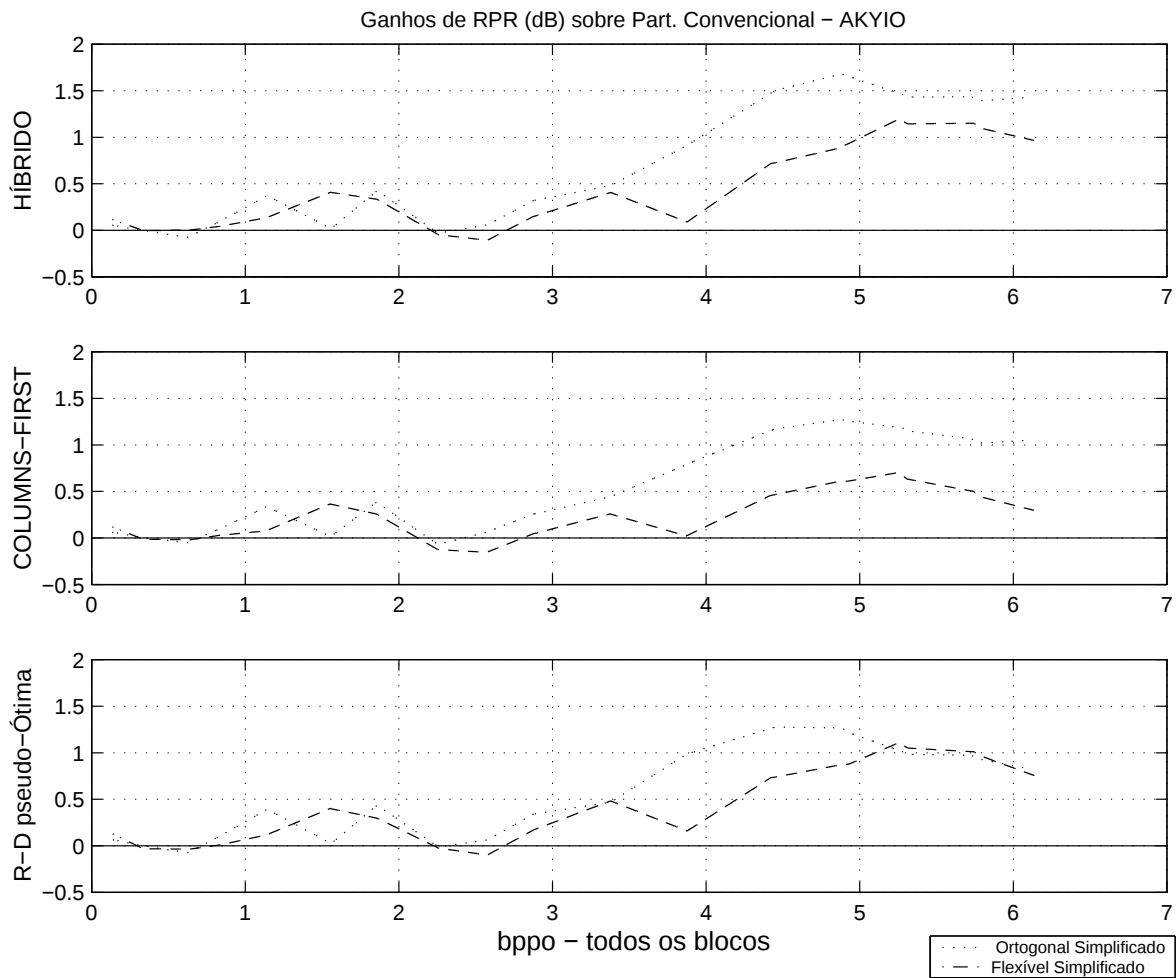


Figura 6.32: Estratégias ‘HÍBRIDA’, ‘COLUMNS-FIRST’ e ‘R-D pseudo-ÓTIMA’: ganhos de RPR (dB) sobre a partição convencional em *AKYIO*

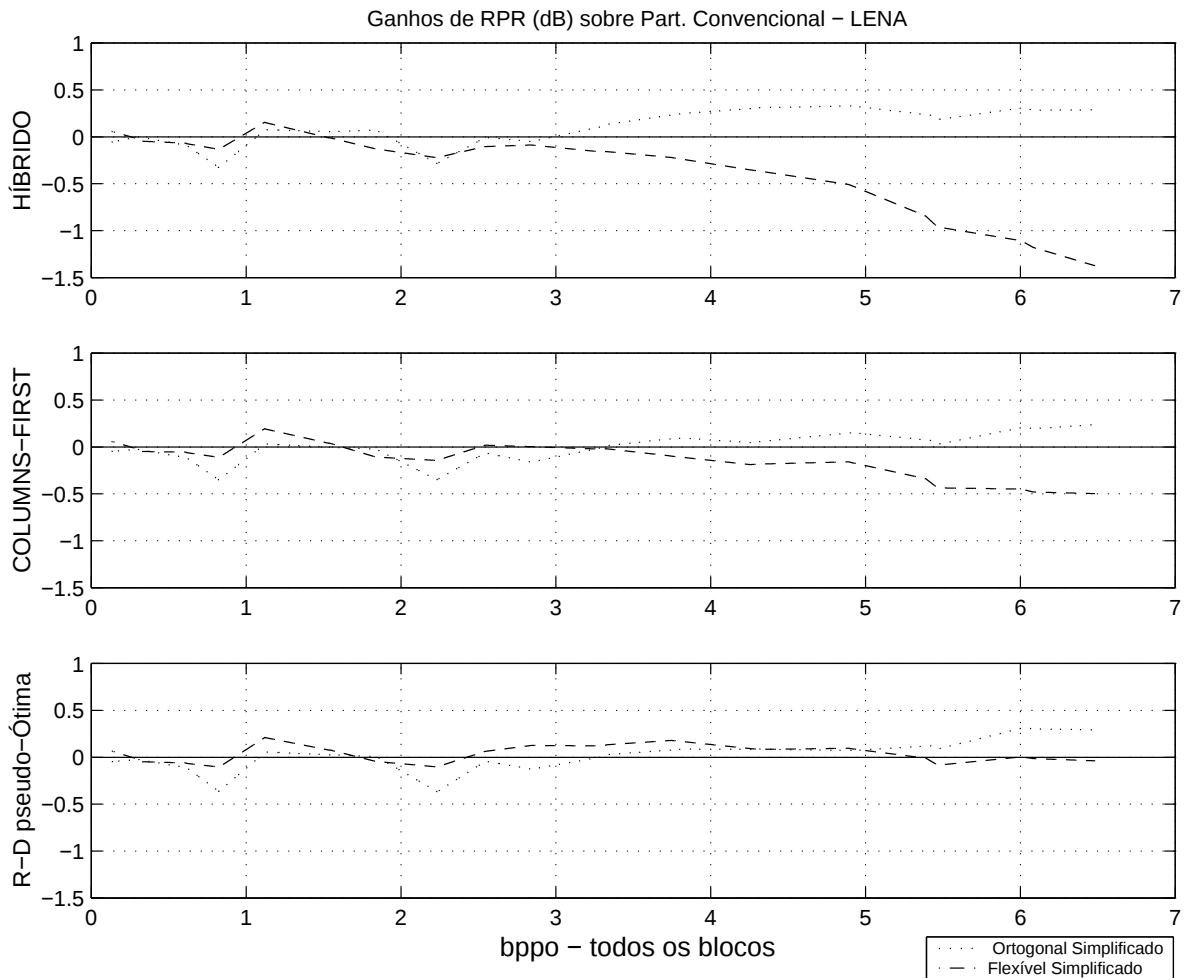


Figura 6.33: Estratégias ‘HÍBRIDA’, ‘COLUMNS-FIRST’ e ‘R-D pseudo-ÓTIMA’: ganhos de RPR (dB) sobre a partição convencional em *LENA*

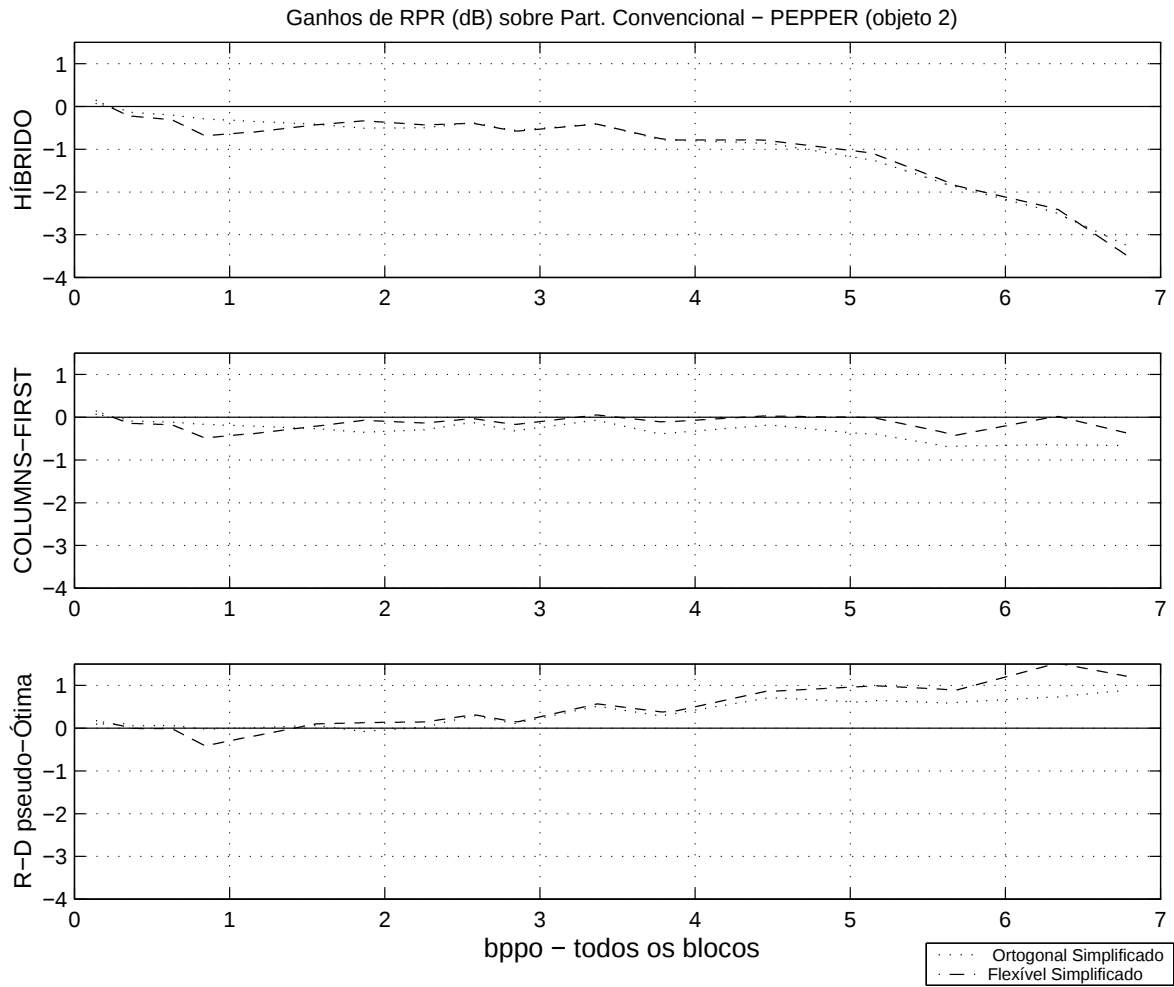


Figura 6.34: Estratégias ‘HÍBRIDA’, ‘COLUMNS-FIRST’ e ‘R-D pseudo-ÓTIMA’: ganhos de RPR (dB) sobre a partição convencional em *PEPPER*(objeto 2)

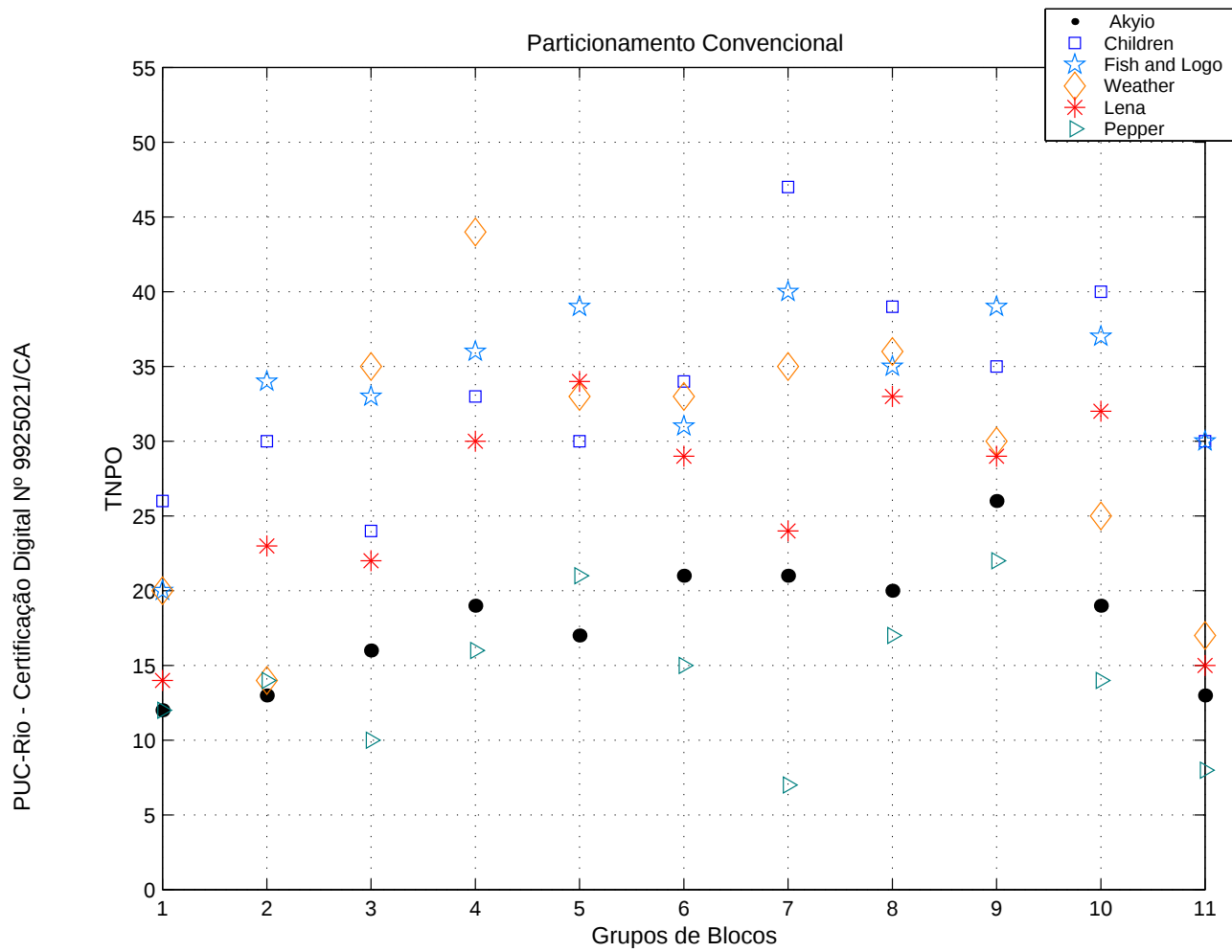


Figura 6.35: Indicador TNPO para o particionamento convencional

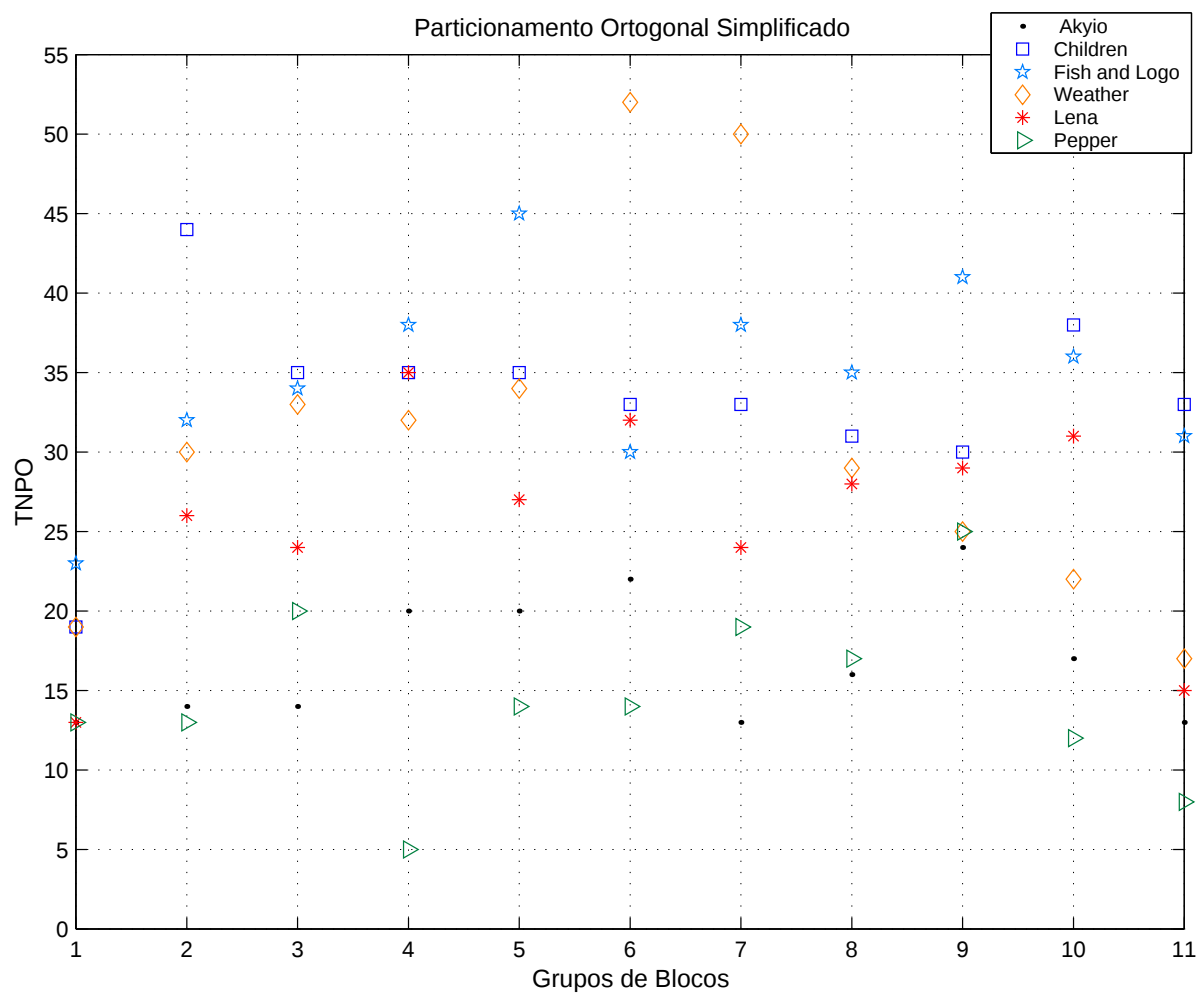


Figura 6.36: Indicador TNPO para o particionamento Ortogonal Simplificado

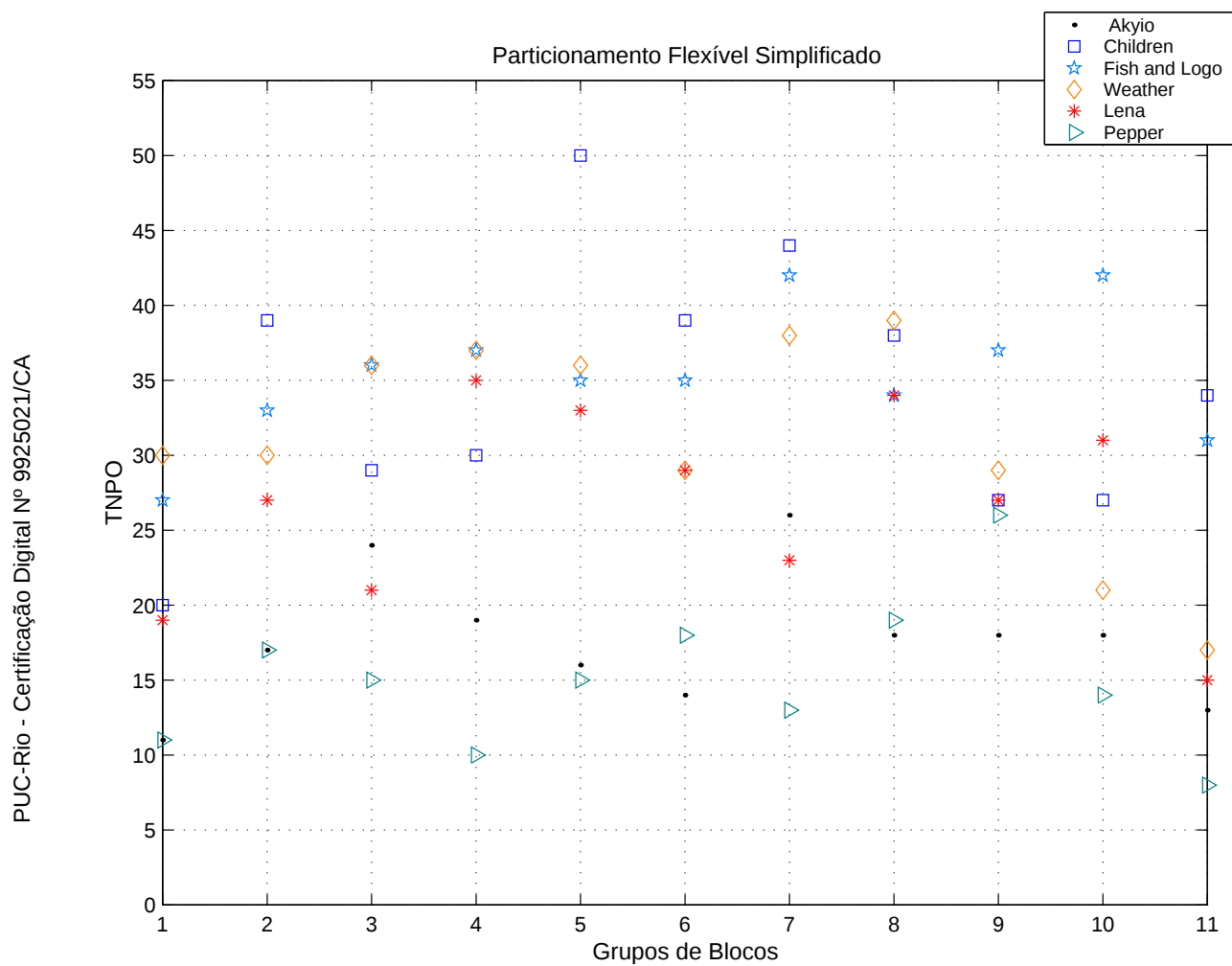


Figura 6.37: Indicador TNPO para o particionamento Flexível Simplificado

No caso do esquema híbrido, apesar de ele ser mais eficiente que MILV e MALV nos particionamentos Convencional e Ortogonal, e de ele ser equivalente a MILV e MALV com o particionamento Flexível, como mostram as Figuras 6.38 a 6.40, a alteração do indicador de textura TNPO de acordo com o método de partição empregado pode levar um grupo de blocos específico a utilizar estratégias distintas para a seleção da direção preferencial (escolhendo entre MILV, MALV e MACES), conforme o esquema de partição adotado. Isso também pode contribuir para a incerteza do desempenho final de codificação.

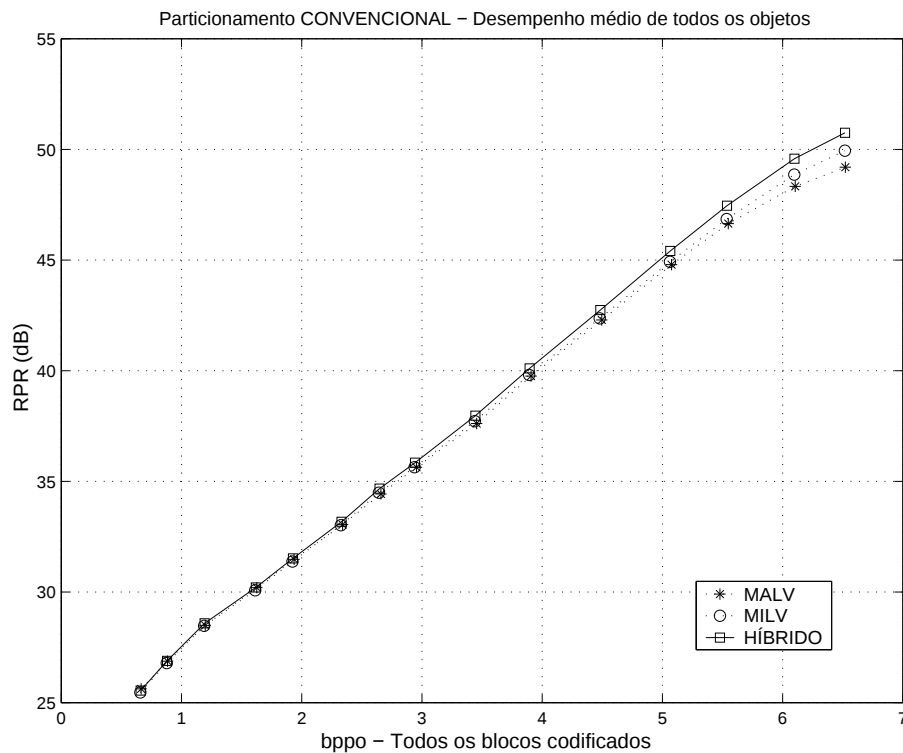


Figura 6.38: Comparação entre as estratégias MILV, MALV e esquema HÍBRIDO com o particionamento Convencional

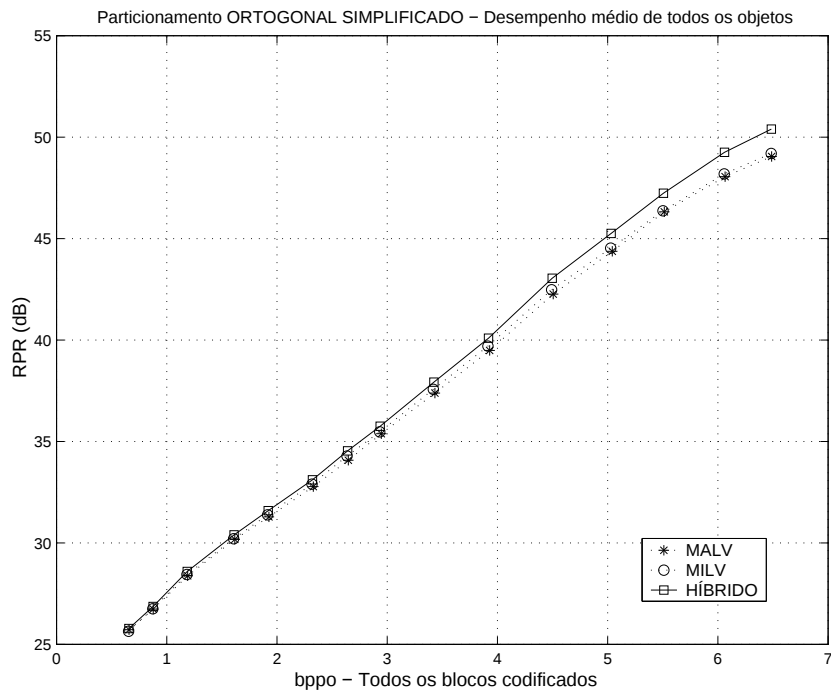


Figura 6.39: Comparação entre as estratégias MILV, MALV e esquema HÍBRIDO com o particionamento Ortogonal Simplificado

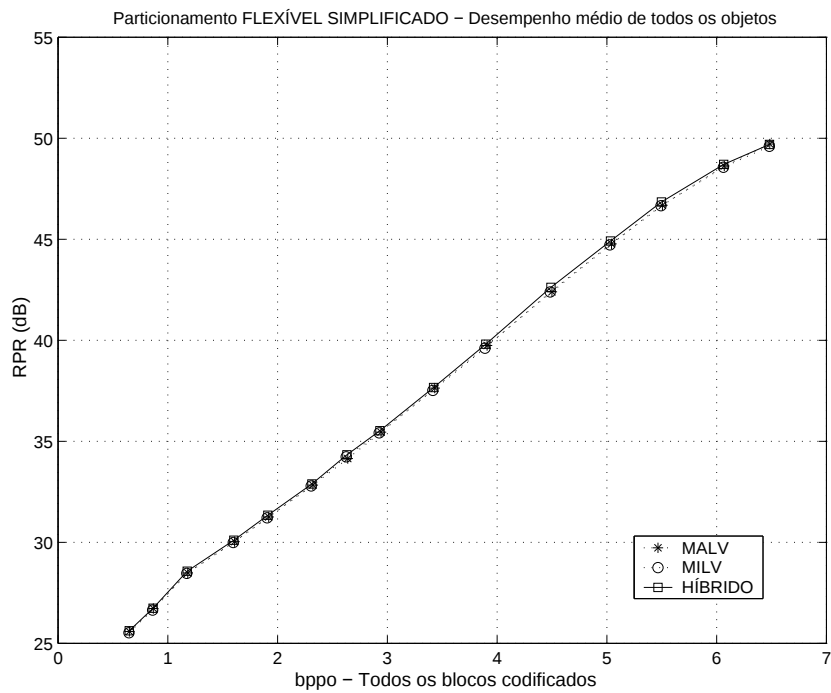


Figura 6.40: Comparação entre as estratégias MILV, MALV e esquema HÍBRIDO com o particionamento Flexível Simplificado

Para explicar a razão da ineficiência dos esquemas de particionamento quando associados à SA-DCT, escolheu-se o objeto *CHILDREN* (objeto 1, quadros 1 a 11) para um estudo mais detalhado do que ocorre com o esquema de codificação HÍBRIDO utilizando os algoritmos de particionamento alternativos. Inicialmente, analisou-se a variação dos indicadores morfológicos ‘DNPO’ - Distribuição percentual dos blocos de acordo com o Número de Pixels do Objeto - e ‘TNPO’ - Textura de acordo com o Número de Pixels do Objeto - ao se empregarem os particionamentos convencional, Ortogonal Simplificado e Flexível Simplificado. A Figura 6.41 mostra o indicador DNPO, ao passo que a Figura 6.42 mostra o indicador TNPO, ambos referentes ao objeto *CHILDREN*.

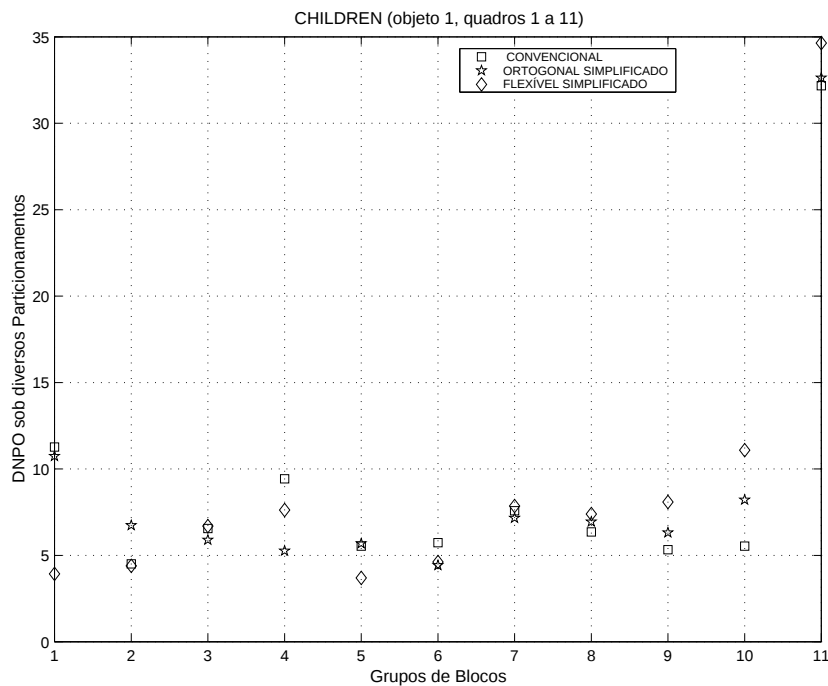


Figura 6.41: Variação do Indicador Morfológico DNPO com o tipo de Particionamento

Analisando a distribuição percentual dos blocos sobre cada um dos grupos de blocos de contorno ($G1$ a $G10$) e de blocos cheios ($G11$) - em relação ao total de blocos a codificar - nos particionamentos convencional, Ortogonal Simplificado e Flexível Simplificado (o que é informado pelo indicador DNPO na Figura 6.41), observa-se que os particionamentos Ortogonal e Flexível proporcionaram um aumento da porcentagem de blocos quase cheios a cheios - $G8$ a $G11$, em relação à partição convencional. Esse aumento foi mais significativo no caso do algoritmo Flexível. A porcentagem dos blocos menos cheios - $G1$ a $G6$ -, ao contrário, diminuiu em relação à

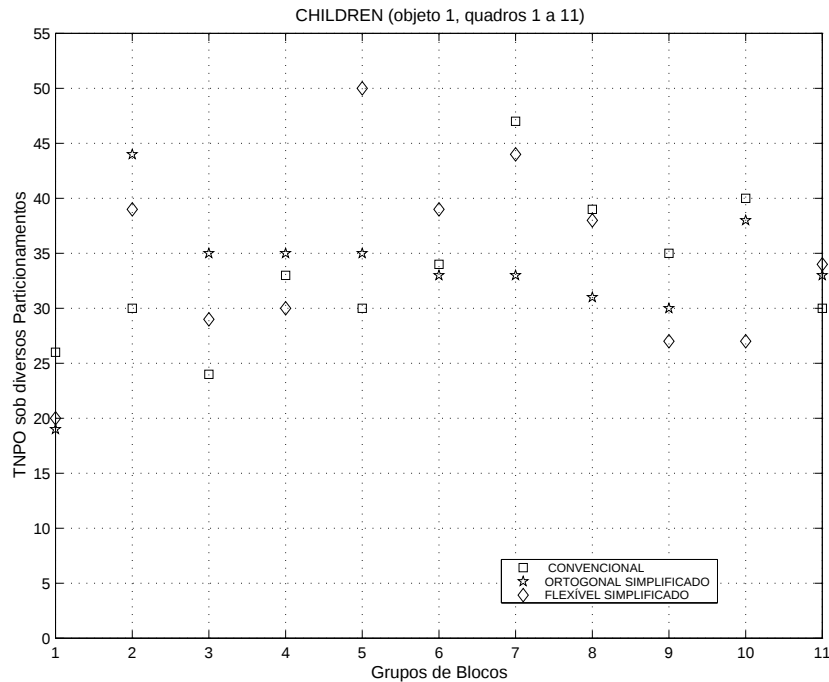


Figura 6.42: Variação do Indicador Morfológico TNPO com o tipo de Particionamento

partição convencional. A exceção ocorreu apenas no grupo $G2$, no qual o algoritmo Flexível proporcionou um aumento percentual do número de blocos, se comparado ao particionamento convencional. Em $G7$, praticamente não houve alteração no número percentual de blocos ao se usar quaisquer dos dois particionamentos alternativos, em relação à partição convencional.

A Figura 6.42, que mostra o indicador TNPO nos três particionamentos, será analisada em conjunto com as Figuras 6.43 a 6.45, que mostram o desempenho de cada um dos grupos de *CHILDREN* quando escolhidos os particionamentos convencional, Ortogonal Simplificado e Flexível Simplificado, respectivamente.

Analisando o grupo $G11$, nota-se que o algoritmo Flexível aumentou o percentual da quantidade de blocos nesse grupo em cerca de 2% do total, enquanto que, no algoritmo Ortogonal, esse percentual quase não se modificou em relação à partição convencional. Contudo, a textura em $G11$ - $TNPO(11)$ - cresceu, se comparada à partição convencional. Esse incremento de textura provocou uma ligeira redução de ganho médio nesse grupo, comparando os algoritmos Ortogonal e Flexível ao método convencional de partição (isso foi mais perceptível no caso do algoritmo Flexível). Portanto, apesar do percentual de blocos em $G11$ ter aumentado, a eficiência de codificação média desses blocos diminuiu.

Análises como essa podem ser feitas para cada um dos grupos, mas as conclusões não são tão imediatas em algumas situações. Em $G1$, tanto o algoritmo Ortogonal quanto o Flexível apresentaram um ganho médio de codificação maior que a partição convencional. Contudo, a porcentagem de blocos nesse grupo é menor em ambos os casos, sendo sensivelmente menor com o algoritmo Flexível. Em geral, os grupos de $G2$ a $G5$ sofreram um decréscimo no ganho de codificação tanto no algoritmo Ortogonal quanto no Flexível, ficando o maior prejuízo por conta de $G5$ no segundo caso, onde os blocos alcançaram uma textura média altíssima. Nos grupos de $G6$ a $G10$, o algoritmo Ortogonal apresentou maiores ganhos de codificação que a partição convencional, havendo as texturas diminuído em todos esses grupos. Daí a razão de, em média, o algoritmo Ortogonal ter se mostrado um pouco mais eficiente que a partição convencional (ver gráfico 1 da Figura 6.29). Já nos grupos $G6$ a $G10$ com o algoritmo Flexível, o ganho de codificação foi reduzido principalmente a taxas altas, apesar de as texturas nos grupos $G7$ a $G10$ também serem menores que na partição convencional. Para explicar o ocorrido, deve-se recorrer à Tabela 6.1, onde foi descrito o esquema híbrido proposto. Observa-se que nos grupos $G9$ e $G10$, a textura média dos blocos com o algoritmo Flexível caiu para valores abaixo de 30 e 35, respectivamente (o que não aconteceu no caso do algoritmo Ortogonal). Com isso, ao invés de ser escolhida a estratégia MILV para a determinação da direção preferencial de processamento da SA-DCT, o esquema híbrido fez a opção pela estratégia MALV, o que, para esses casos, foi menos eficiente. Lembrando que o percentual de blocos em $G9$ e $G10$ aumentou com os algoritmos Ortogonal e Flexível (ver indicador DNPO na Figura 6.41), o ganho final de codificação tendeu a aumentar com o algoritmo Ortogonal (para algumas taxas de bits) e a reduzir com o algoritmo Flexível (em praticamente todas as taxas de bits). Já nos grupos $G7$ e $G8$, tanto o algoritmo Ortogonal quanto o Flexível produziram texturas que levaram a estratégia MILV ser escolhida pelo esquema híbrido. Como as texturas nesses grupos são menores para o algoritmo Ortogonal, os ganhos médios de codificação foram maiores para esse algoritmo que para o algoritmo Flexível.

Os resultados de codificação dos demais objetos e das outras estratégias para a determinação da primeira direção de processamento da SA-DCT podem ser analisados de forma semelhante, levando à conclusão de que a utilização dos esquemas de particionamento alternativos não é eficiente para a melhoria da eficiência de codificação da SA-DCT. Isso é verdade para qualquer estratégia usada para a determinação da direção preferencial de processamento da SA-DCT, inclusive a ‘R-D pseudo-ÓTIMA’.

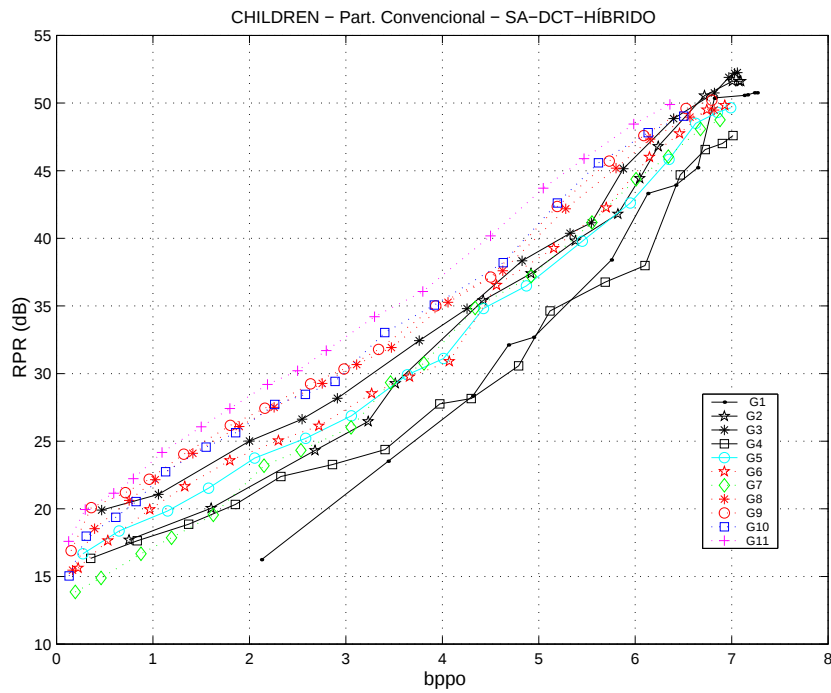


Figura 6.43: Desempenho dos grupos de *CHILDREN* quando empregado o particionamento convencional

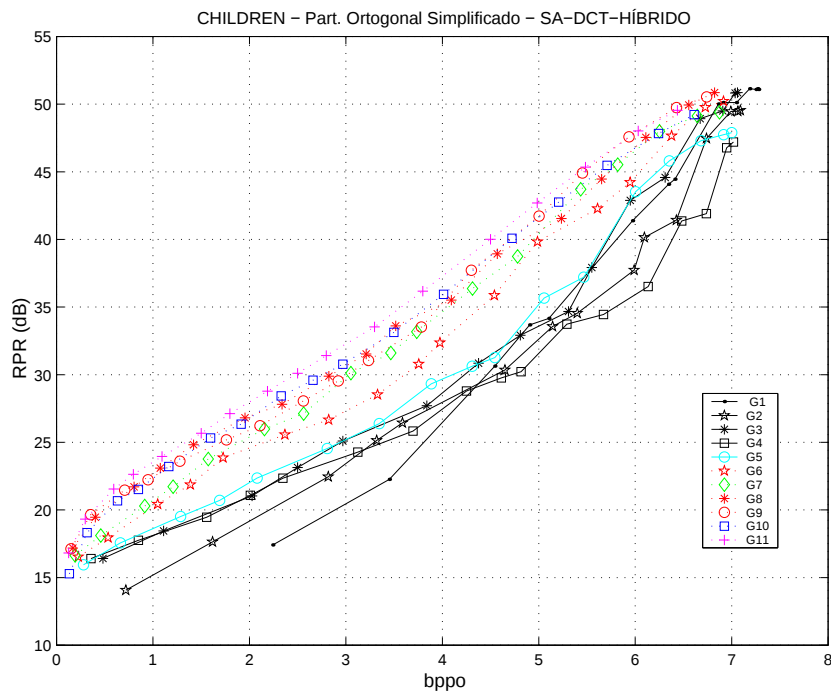


Figura 6.44: Desempenho dos grupos de *CHILDREN* quando empregado o particionamento Ortogonal Simplificado

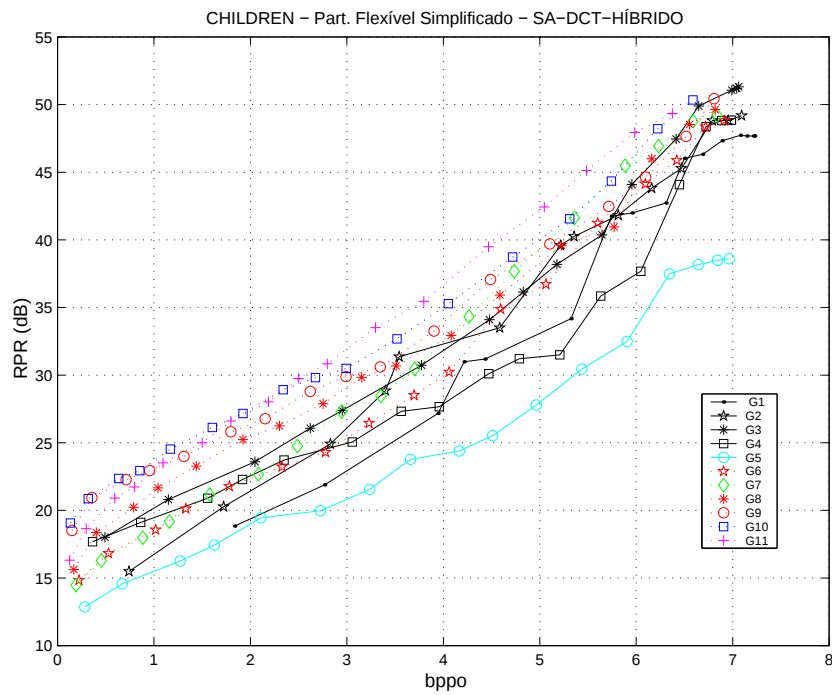


Figura 6.45: Desempenho dos grupos de *CHILDREN* quando empregado o particionamento Flexível Simplificado

6.5

Conclusões

Este capítulo analisou a eficiência das estratégias MALV e MILV para a determinação da direção prioritária de processamento da SA-DCT, em aplicações de codificação de imagem orientada por objeto. Foi mostrado que nem sempre MALV é superior a MILV, como reportado na literatura [77].

Neste capítulo, também foi apresentada uma avaliação criteriosa sobre as variações dos desempenhos de MILV e MALV sobre os grupos de blocos de contorno, de acordo com o indicador morfológico TNPO. Os resultados dessa análise permitiram a seleção da estratégia mais adequada a cada grupo, apontando também para algumas condições em que a relação entre o indicador TNPO e as eficiências de MILV e MALV é comprometida. Em seguida, o indicador TNPO foi aplicado como a base de um esquema híbrido que seleciona adaptativamente a melhor estratégia para um grupo específico de blocos de contorno. Preferencialmente, a estratégia selecionada é MALV ou MILV, mas há casos em que a opção é pela estratégia MACES, que mede a capacidade de compactação de energia das transformadas.

Foi mostrado que a eficiência do esquema híbrido proposto é comparável ou superior ao das estratégias MALV, MILV e ‘COLUMNS-FIRST’ (algoritmo original de Sikora [67]). Ressalta-se que a escolha da direção preferencial para um bloco específico independe da taxa de bits requerida, o que é uma vantagem sobre o critério que escolhe a direção que produz o menor produto $emq \times nbits$ [77]. Além disso, a complexidade computacional do esquema híbrido é muito menor, uma vez que não requer a codificação de dois blocos para cada bloco de contorno (tomando-se ora a direção vertical como prioritária, ora a direção horizontal), e muito menos a recuperação desses blocos codificados para a escolha da direção preferencial de processamento em um bloco de contorno específico.

Finalmente, foram apresentados os ganhos de codificação obtidos com o emprego dos algoritmos de particionamento de blocos alternativos à partição convencional [76],[94]. Esses resultados foram mostrados para as estratégias ‘COLUMNS-FIRST’, ‘HÍBRIDA’ e ‘R-D pseudo-ÓTIMA’, que determinam a direção prioritária de processamento da SA-DCT. Foi verificado que não existe garantia de que esses algoritmos possam melhorar a eficiência de codificação ao concentrar os pixels do objeto em um menor número de blocos, como havia sido afirmado em [76]. Na verdade, o ganho de codificação pode inclusive diminuir. Dado ainda que o emprego desses algoritmos implica em aumento da carga computacional tanto do codificador

quanto do decodificador, o seu uso não é, de forma alguma, indicado no caso da codificação adaptativa à forma (SA-DCT), qualquer que seja a estratégia usada para a determinação da direção preferencial de processamento.

Contudo, ressalta-se que o algoritmo híbrido proposto manteve-se superior ou equivalente em desempenho às estratégias MILV e MALV ao utilizarem-se todos os esquemas de particionamento, o que mostra a eficácia da utilização dos Indicadores Morfológicos para a escolha da estratégia mais eficiente (entre MILV, MALV e MACES) para a determinação da direção preferencial de processamento da SA-DCT.