

5 Experimentos e Resultados

Neste capítulo serão apresentados os resultados obtidos no desenvolvimento deste trabalho. Para cada resultado apresentado, pretende-se fazer uma breve discussão sobre os mesmos.

Para a compressão dos dados sísmicos foram utilizadas duas combinações de algoritmos. A primeira combinação foi o uso do algoritmo de Huffman seguido do algoritmo de RLE. A segunda combinação é o inverso da primeira. Os resultados comparativos são apresentados na seção 5.1.

Na seção 4.1.1 foi proposto o uso da transformada *wavelet* 3D, no lugar da transformada *wavelet* 2D, para compressão de dados sísmicos volumétricos. Logo, este capítulo apresentará uma comparação entre a compressão de dados utilizando estas duas transformadas. Anteriormente, foi definido que neste trabalho seria usada a transformada de Haar, e que esta tem a limitação de que o dado sísmico tem que ter suas dimensões como sendo potência de dois. Serão apresentados testes de compressão com dados que não possuem estas características, portanto será necessário aplicar um pré-processamento a estes dados. Estes resultados são apresentados na seção 5.2.

Por fim, foi sugerida a introdução do traço complexo ao processo de compressão de dados sísmicos volumétricos. Na seção 5.3, são apresentados os resultados obtidos com este novo método.

Os dados sísmicos utilizados neste capítulo são provenientes do projeto *OpendTect* [41]. Os dados sísmicos *Volume1*, *Volume2* e *Volume3* são um recorte do dado sísmico *F3_Block*, disponibilizado no projeto *OpendTect*, com dimensões de $256 \times 256 \times 256$, de diferentes regiões do dado original. Os dados *nãoCubo1* e *nãoCubo2* também são recortes do dado *F3_Block*, porém suas dimensões não são iguais.

As taxas de compressão são apresentadas pela notação $X:1$, onde X representa a razão entre o tamanho do dado comprimido pelo tamanho do dado original. Esta notação foi escolhida para permitir que os resultados encontrados

possam ser comparados com outros trabalhos presentes na literatura, visto que esta notação é largamente utilizada. O erro obtido na reconstrução do dado original é apresentado utilizando o SNR pelo mesmo motivo do uso da taxa de compressão.

5.1. Ordem dos algoritmos de compressão

No trabalho de Averbuch *et al* [3], foi descrito o uso do algoritmo de Huffman seguido pelo RLE para compressão do dado sísmico. A idéia é que o algoritmo de Huffman cria um dicionário formado por zeros e uns para representar todos os símbolos existentes no dado sísmico. Estes zeros e uns eventualmente se repetem e, portanto, justifica-se o uso do RLE.

Deve-se tomar um cuidado especial com a ordem em que os algoritmos são aplicados. É mais vantajoso aplicar primeiro o RLE e em seguida o Huffman quando o dado sísmico é pequeno, e este está em memória. Porém, caso o dado não esteja em memória é melhor primeiro aplicar o Huffman e em seguida o RLE.

Para ilustrar este problema, será apresentado um exemplo de compressão utilizando as duas possibilidades de aplicação dos algoritmos. Para codificar a mensagem {1, 1, 1, 3, 2, 6, 6, 2, 8, 8, 8, 8, 1, 3, 3, 5, 5, 5, 5, 4}, utilizando a codificação mínima, conforme descrito na seção 2.4.1, seriam necessários 160 *bits* para representar esta mensagem. Caso fosse utilizado o algoritmo de Huffman e, em seguida, RLE o resultado seria uma mensagem de 225 *bits*, ou seja, pior que a codificação mínima. Este valor refere-se apenas à mensagem. Não leva em consideração a informação extra da árvore de Huffman para a descompressão do dado. No caso de ser aplicado primeiro o algoritmo de RLE e em seguida o Huffman, o resultado é uma representação de 51 *bits* para a mensagem, também sem levar em consideração a árvore de Huffman.

Porém, em se tratando de arquivos sísmicos, muitas vezes estes dados não cabem em memória. Por isso, foram utilizados algoritmos de Huffman e de RLE que são aplicados diretamente ao arquivo. Nestes algoritmos é lida uma amostra, um caractere do tipo *unsigned char*, e feito o processamento desta amostra. Neste caso, a aplicação do algoritmo de Huffman antes do algoritmo RLE mostrou-se mais vantajoso.

O gráfico 3 mostra um comparativo entre a aplicação das duas combinações ao dado sísmico *Volume1*. Neste gráfico foi utilizada a técnica descrita de ler amostra por amostra e processar a amostra lida. Percebe-se que o uso de Huffman seguido do RLE é mais vantajoso.

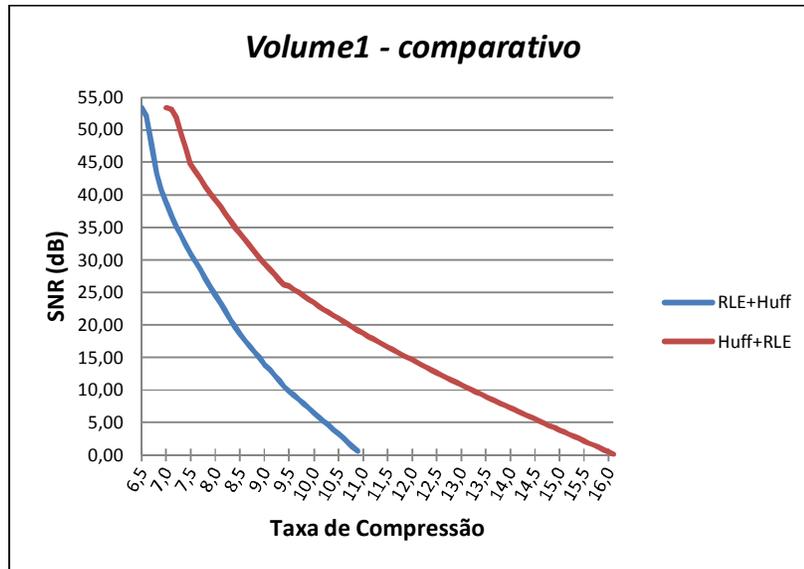


Gráfico 3 Comparativo entre a ordem de execução dos algoritmos de Huffman e RLE no dado sísmico *Volume1*.

No gráfico 4 tem-se a mesma comparação para outro dado sísmico. Mais uma vez, é melhor primeiro aplicar o Huffman e em seguida aplicar o RLE. O mesmo ocorreu para os outros dados utilizados ao longo deste trabalho.

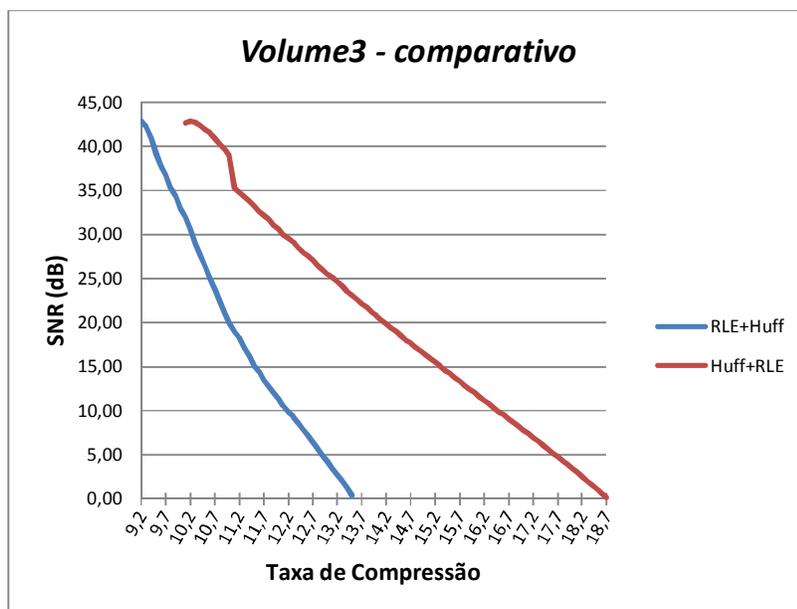


Gráfico 4 Comparativo entre a ordem de execução dos algoritmos de Huffman e RLE no dado sísmico *Volume3*.

5.2.

Comparativo: *Wavelet 2D* x *Wavelet 3D*

Foram geradas curvas de compressão dos dados sísmicos *Volume1*, *Volume2* e *Volume3* utilizando a transformada *wavelet 3D* e a transformada *wavelet 2D*. Os resultados são analisados a seguir.

Para gerar a curva que representa a decomposição do eixo *crossline*, foi feita para cada seção *crossline* do dado, a decomposição *wavelet 2D* desta seção. Ou seja, para cada seção *crossline* é feita a decomposição no eixo *inline* e tempo. A figura 37 mostra algumas seções *crossline* decompostas.

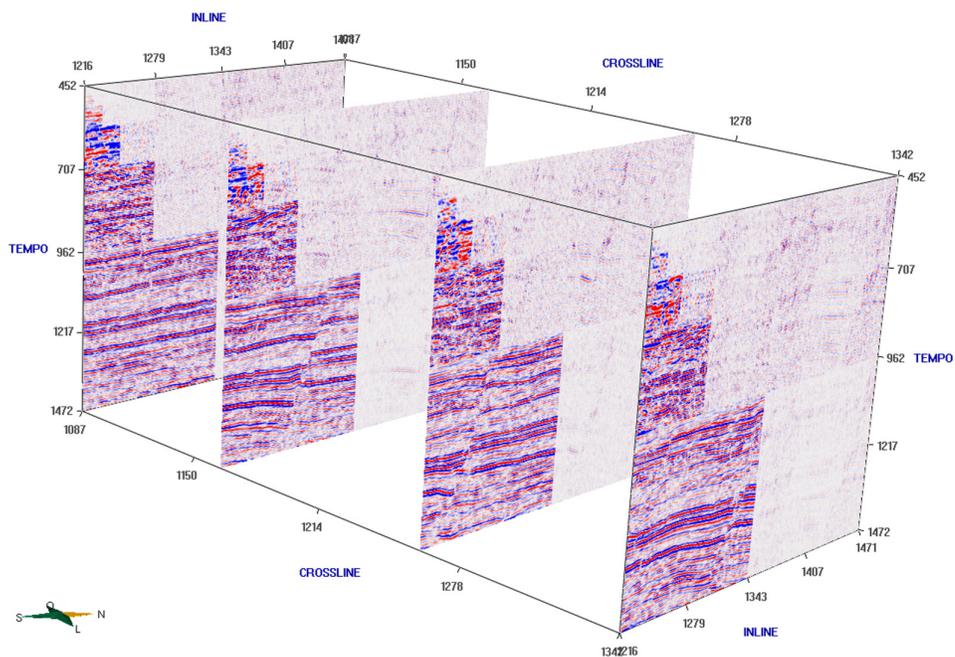


Figura 37 *Volume1* decomposto utilizando a transformada *wavelet 2D* nas seções de *crossline*.

O gráfico 5 mostra a comparação das curvas de compressão entre a transformada *wavelet 3D* e a transformada *wavelet 2D* nos eixos *inline* e *crossline*.

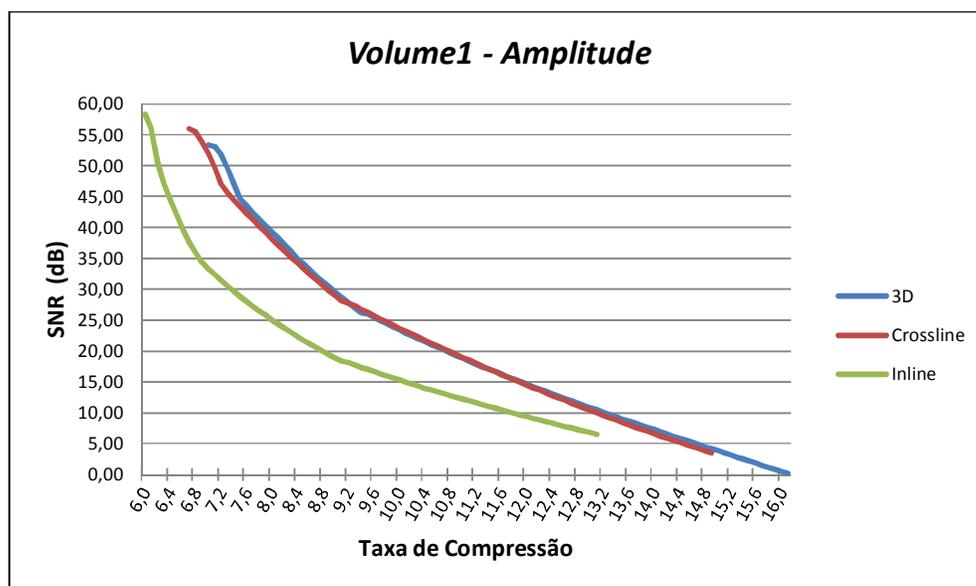


Gráfico 5 Curvas de compressão para o dado sísmico Volume1.

É justificável o fato da curva de compressão da *crossline* ser melhor que a curva da *inline*, pois, em geral, a aquisição do dado sísmico é feita de forma que o eixo *inline* represente o eixo de “mergulho” das camadas sísmicas. Com isso, as seções *crossline* tendem a representar uma maior coerência lateral no dado sísmico. A transformada *wavelet* 3D leva vantagem em relação a 2D, pois a cada nível de decomposição é feita a transformada nas três direções, mantendo a coerência do dado original.

Estes gráficos, onde são apresentadas as curvas de compressão de um determinado atributo sísmico, são gerados variando um único parâmetro, que é o tamanho do intervalo que contém o zero na quantização. Ao variar este parâmetro, a taxa de compressão obtida também irá variar. Porém, em alguns casos, se o passo de discretização for muito pequeno, a taxa de compressão pode não variar, mas o SNR obtido irá variar. Nestes casos, é escolhido o maior SNR encontrado para representar a taxa de compressão correspondente.

No gráfico 5, percebe-se que a curva de compressão da transformada *wavelet* 3D está muito próxima da curva de compressão da *crossline*. Em alguns pontos o erro introduzido no dado sísmico foi menor na transformada 3D, e em outros pontos, o erro da transformada 2D foi menor.

O gráfico 6 mostra a mesma comparação para o dado sísmico *Volume2*. Neste dado, a transformada *wavelet* 3D teve um desempenho melhor que a transformada 2D. Não só a taxa de compressão foi maior como o SNR também

foi melhor, ou seja, para qualquer taxa de compressão obtida pela transformada 2D, a 3D obteve a mesma taxa de compressão com uma menor introdução de ruído ao dado original.

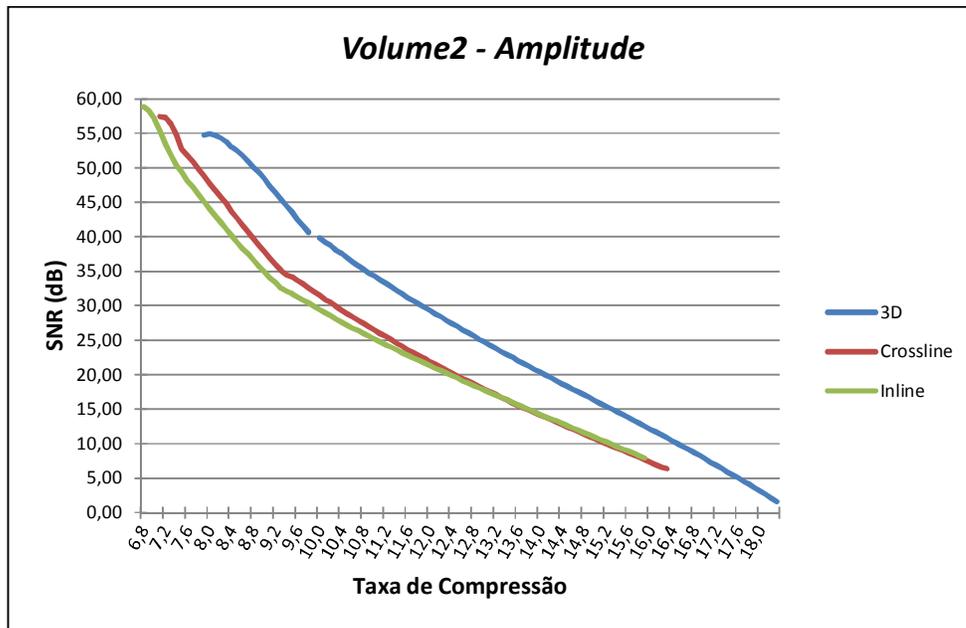


Gráfico 6 Curvas de compressão para o dado sísmico Volume2.

Os gráficos apresentados com as curvas de compressão, não são funções contínuas, são um conjunto de pontos que quando os parâmetros de discretização do Z é muito denso, forma uma linha mostrando o comportamento da compressão dos mesmos. A descontinuidade apresentada na curva de compressão da transformada *wavelet* 3D, se deu devido ao passo da variável Z ser maior que o necessário para englobar todos os valores da taxa de compressão. Se o intervalo de incremento do Z fosse menor que todas as taxas de compressão, os pontos do eixo horizontal do gráfico, teriam um SNR associado.

No gráfico vê-se que o comportamento das curvas de compressão são semelhantes ao apresentado no gráfico 5. A transformada *wavelet* 3D tem o comportamento próximo ao da curva da *crossline*.

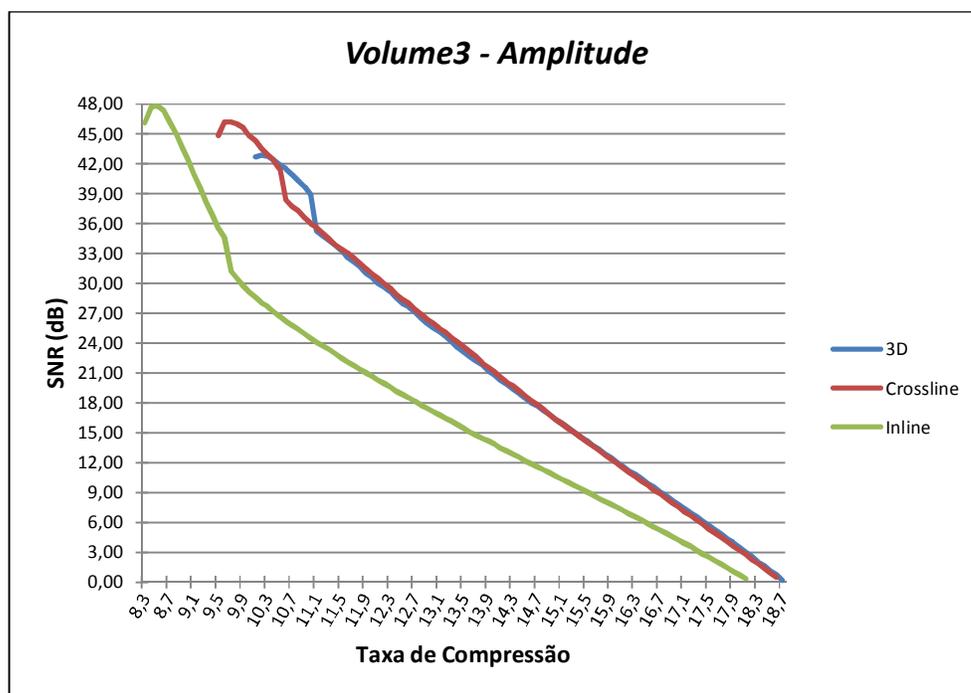


Gráfico 7 Curvas de compressão para o dado sísmico Volume3.

Observando os gráficos das curvas de compressão para os dados sísmicos *Volume1*, *Volume2* e *Volume3*, percebe-se que a curva de compressão para a *crossline* e a curva para a *inline* têm comportamentos diferentes nos gráficos apresentados. Isso levou a pergunta de qual seria o comportamento da *wavelet 3D* se a ordem das transformações fosse alterada.

Nos gráficos apresentados, a transformada *wavelet 3D* foi feita primeiro no eixo da *inline*, em seguida no eixo do tempo e por fim no eixo da *crossline*. Foi gerada uma nova decomposição do dado sísmico utilizando a transformada *wavelet 3D*, porém a decomposição foi aplicada primeiro no eixo da *crossline*, em seguida no eixo da *inline* e, por último, no eixo do tempo. Um novo dado foi gerado utilizando a decomposição nos eixos tempo, *inline* e *crossline*, respectivamente.

Este experimento foi repetido com todos os dados sísmicos citados. Em todos eles, o resultado foi que a ordem de decomposição dos eixos não altera o resultado final da compressão. Ou seja, o resultado da transformada *wavelet 3D* não varia com a ordenação do dado sísmico, o que não ocorre com a transformada 2D. Como mostra o gráfico, na transformada 2D se a decomposição é aplicada ao eixo *inline* os resultados são diferentes se aplicados ao eixo *crossline*.

Como foi dito anteriormente, uma limitação da *wavelet* de Haar é que o dado seja potência de dois, como neste trabalho estão sendo utilizados dados sísmicos volumétricos, é necessário que as dimensões do arquivo sejam potência de dois. Os quatro dados apresentados anteriormente tinham essa característica. A seguir, será apresentado um dado sísmico que não possui tais características e a comparação entre a transformada *wavelet* 3D e 2D.

O dado sísmico *nãoCubo1* é um dado sísmico que possui 215 amostras no eixo *inline*, 170 amostras no eixo *crossline* e 230 amostras no tempo. Ao ser feita a transformada de Haar 3D neste dado, o mesmo será completado com zeros de forma que as dimensões do dado fiquem $256 \times 256 \times 256$.

Este dado possui o tamanho em disco de 32,1 MB. Ao completar o dado sísmico com zeros, para fazer a transformada *wavelet* 3D, este dado passa a ocupar um espaço em disco de 64 MB. A taxa de compressão é calculada sempre sobre o dado original, ou seja, sobre os 32,1 MB originais.

No caso da transformada *wavelet* 2D, quando feita sobre o eixo da *crossline*, é necessário transformar o eixo da *inline* e do tempo em potências de dois. Analogamente, quando a transformada é feita no eixo *inline*, completam-se os eixos *crossline* e tempo.

No caso do dado não ser potência de dois, e se for utilizada a transformada *wavelet* 2D, é mais vantajoso fazer a transformada no menor eixo, pois assim, ao completar o dado, serão introduzidas menos amostras ao dado. Por exemplo, no dado *nãoCubo1* o menor eixo é a *crossline*, com 170 amostras. Logo, para cada seção sísmica deste dado, deverão ser introduzidas 67 amostras na seção, sendo 41 amostras na *inline* e 26 amostras no tempo. Isso dá um total de 11.390 amostras a serem introduzidas ao dado. Este número de amostras introduzidas faz com que o dado passe a ocupar um espaço em disco de 42,5 MB. Se a transformada fosse aplicada sobre o eixo da *crossline* o tamanho do dado em disco passaria para 53,8 MB.

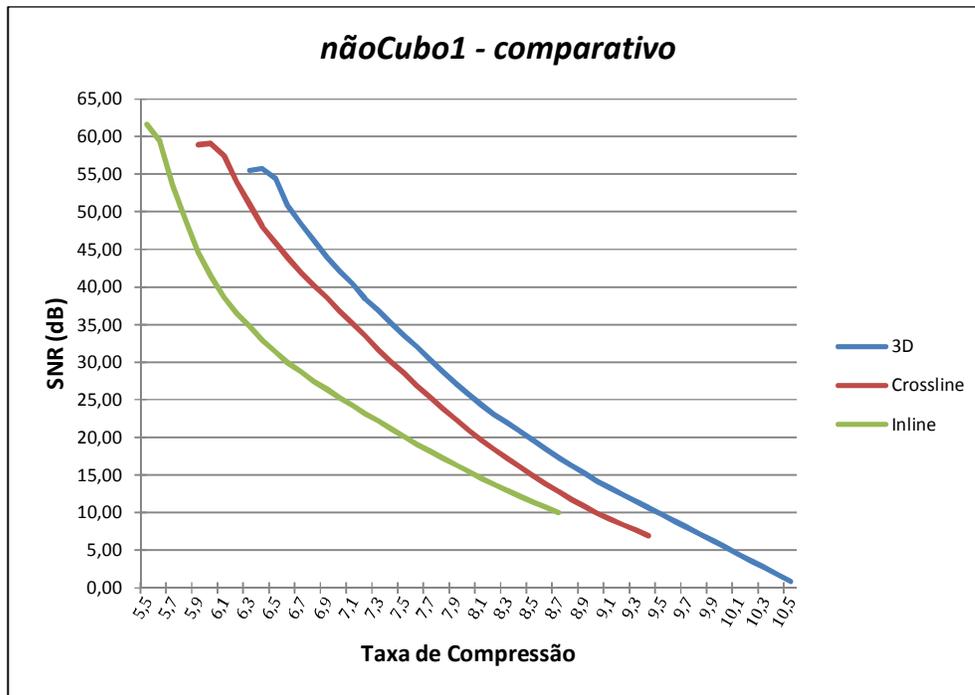


Gráfico 8 Comparativo da transformada wavelet 3D e 2D para dados sísmicos cujas dimensões não são potência de dois.

O gráfico 8 mostra a vantagem da transformada *wavelet* 2D no eixo da *crossline* em relação ao eixo da *inline*. Apesar da transformada 3D ter sido vantajosa em relação a 2D, nos casos dos dados que não sejam potência de dois isto nem sempre acontece.

O próximo dado a ser apresentado é o dado *nãoCubo2*. Este dado possui as mesmas dimensões do dado *nãoCubo1*, sendo a principal diferença entre estes dois dados é que o *nãoCubo2* possui a parte superior zerada, representando uma camada de água, similar aos dados sísmicos utilizados nos trabalhos de Averbuch *et al* [3] e de Wu *et al* [4].

O gráfico 9 mostra as curvas de compressão para as transformadas *wavelet* 3D e 2D.

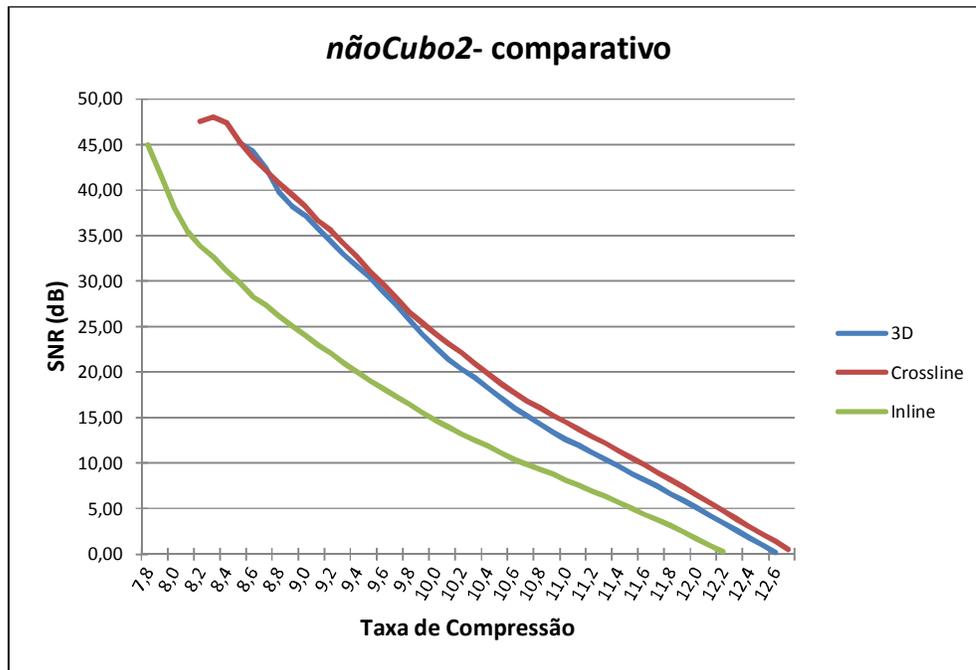


Gráfico 9 Comparativo da transformada wavelet para dado sísmico não cubo e com fundo do Mar.

Ao comparar as taxas de compressão apresentadas no gráfico 9 com as taxas apresentadas no gráfico 8 tem-se uma maior taxa de compressão no dado que possui uma parte zerada. Isto ocorre, pois, o algoritmo de Huffman tira proveito da maior probabilidade de ocorrência do zero.

5.3. Separar em Atributos

Na seção 4.2.2 foi introduzida a idéia de separar a amplitude sísmica em dois atributos e fazer a compressão sobre estes dois atributos. Em seguida, na seção 4.1.3 foi apresentado um gráfico mostrando que a taxa de compressão dos atributos de envelope e de frequência eram melhores que a taxa de compressão da amplitude sísmica.

O gráfico 10 a seguir, mostra que para o dado sísmico *Volume2*, a idéia também é válida.

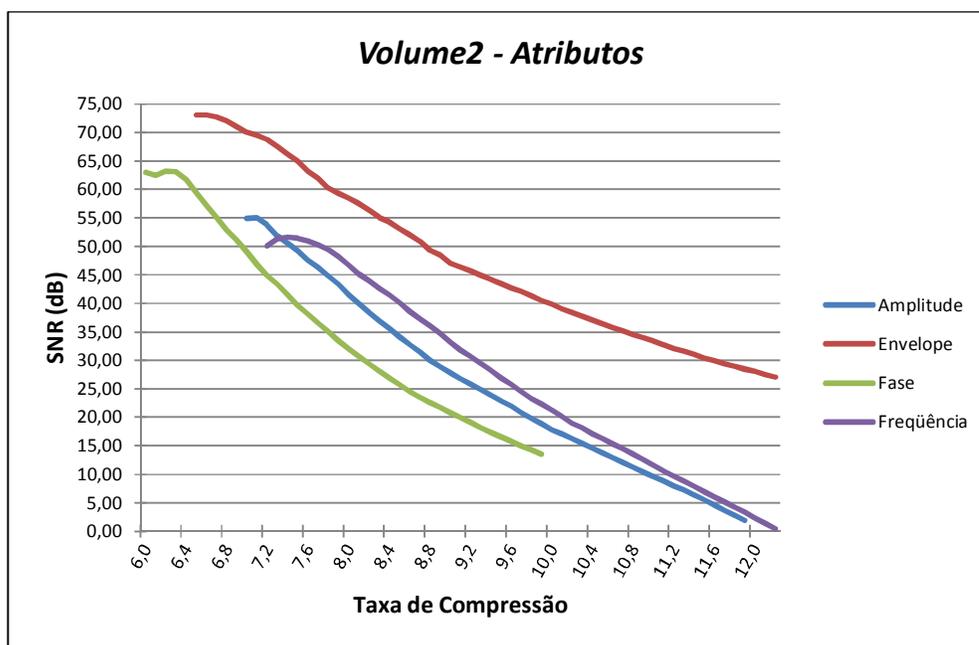


Gráfico 10 Curva de compressão entre os atributos sísmicos do dado Volume2.

Pelas curvas de compressão do gráfico 2 e do gráfico 8, é esperado que a combinação de envelope e frequência instantânea obtenham uma maior taxa de compressão. É importante lembrar que analisar apenas a taxa de compressão não é suficiente. Deve-se atentar ao ruído introduzido ao dado, pois este é que irá determinar se a compressão obtida é válida ou não. Pois, se um dado foi comprimido com a taxa de compressão de 90:1 e obtido o SNR de 2dB, este dado terá tanto ruído que será inviável fazer qualquer tipo de processamento.

Ao dividir o dado sísmico original em outros dois atributos, dobra-se o espaço em disco necessário para armazená-los. Porém, o cálculo da taxa de compressão deve ser feito levando-se em conta o tamanho do dado original. O cálculo do SNR será feito sobre o dado original e o dado reconstruído a partir dos dois atributos.

Como foi demonstrado na seção 4.2.1, já existe a introdução de ruído ao dado sísmico apenas ao fazer a decomposição do dado original nos atributos sísmicos e reconstruir o dado original. E como dito anteriormente, a quantização é responsável pela maior perda de informação existente no processo.

Ao fazer todas as etapas do estudo de separar em atributos, o SNR obtido foi menor que um, ou seja, a quantidade de ruído contido no dado sísmico reconstruído era maior que a quantidade de informação original contida no dado. Foram utilizados os atributos de envelope e fase instantânea, com uma

quantização em um *byte* e com 256 classes. O mesmo ocorreu ao fazer a combinação de envelope e frequência.

Para tentar reduzir a introdução de ruído ao dado sísmico, a quantização foi alterada para dois *bytes* e utilização de 2048 classes. É evidente que esta modificação da quantização irá acarretar na perda da taxa de compressão, porém espera-se que tenha benefícios no SNR resultante. O gráfico 11 mostra a comparação entre as duas quantizações citadas.

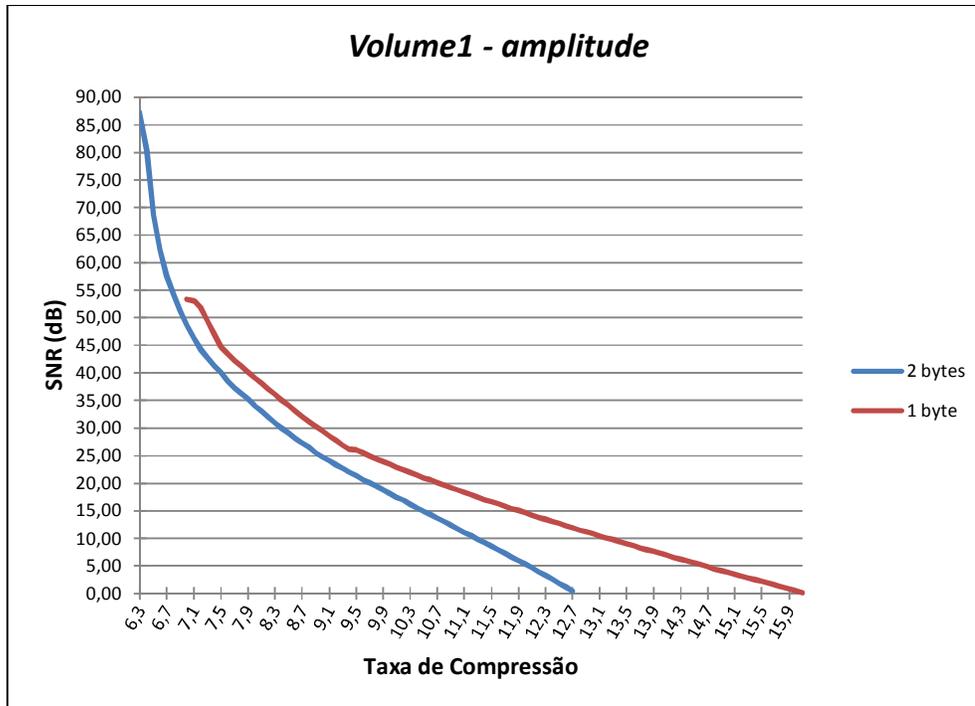


Gráfico 11 Comparativo entre a quantização de 256 classes e de 2048 classes.

Como esperado, o SNR inicial é maior no dado quantizado com dois *bytes*. Porém, como este dado possui um maior número de classes, conseqüentemente, uma menor probabilidade de ocorrência de um determinado símbolo, as taxas de compressão obtidas com o algoritmo de Huffman são piores que na quantização de um *byte*.

No entanto, ao fazer a decomposição da amplitude sísmica em dois atributos, fazer a transformada *wavelet*, quantizar e comprimir os atributos resultantes, utilizando esta quantização de dois *bytes*, é possível aplicar as transformações inversas aos atributos resultantes e reconstruir a amplitude original. O gráfico 12 mostra a taxa de compressão entre as decomposições dos atributos sísmicos do *Volume1*.

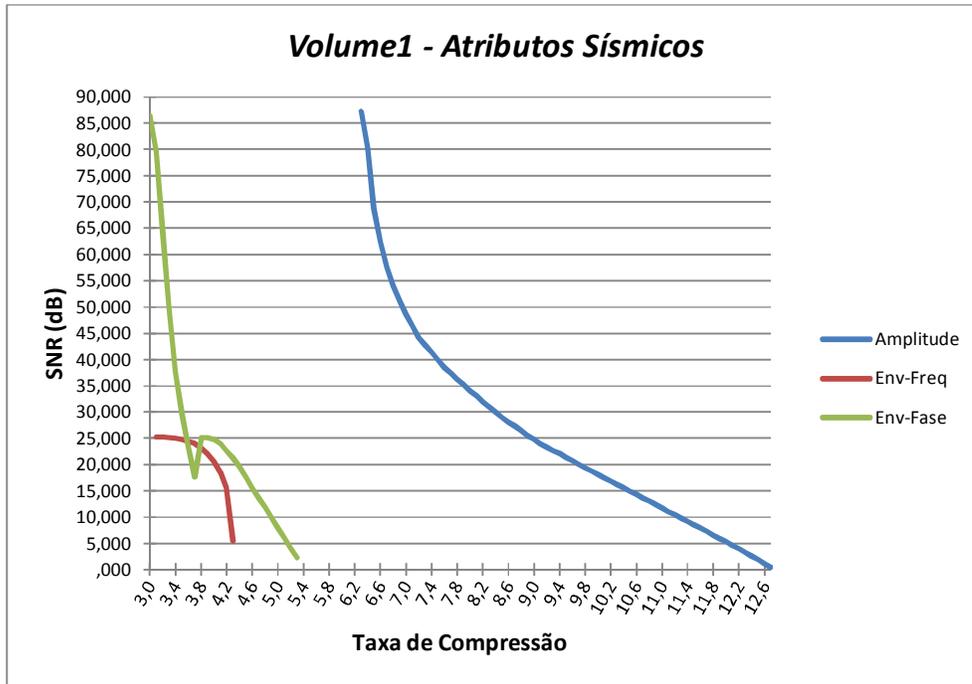


Gráfico 12 Comparativo entre a taxa de compressão da combinação de atributos sísmicos para o volume1.

Foi feito o mesmo teste para o Volume2, utilizando uma quantização de dois bytes e 1024 classes. Mais uma vez, a amplitude obteve uma melhor resultado em relação a decomposição de atributos. O gráfico 13 mostra estes resultados.

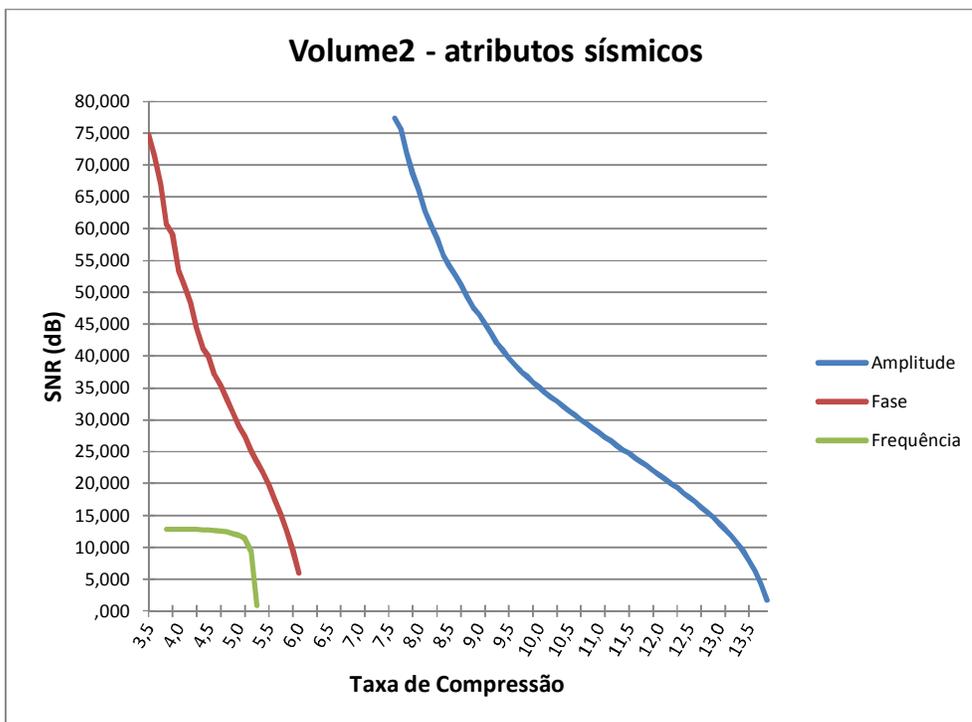


Gráfico 13 Comparativo entre a taxa de compressão da combinação de atributos sísmicos para o volume2.

Ao analisar os gráfico 12 e gráfico 13, percebe-se que o erro introduzido na amplitude sísmica ao fazer a compressão dos atributos sísmicos é muito maior que fazer a compressão diretamente no atributo de amplitude.

Para gerar a curva de compressão com os atributos sísmicos, primeiro fixou-se o tamanho do intervalo que contém o zero de um atributo, quantizou-se este atributo e o comprimiu. Guardado o tamanho do dado gerado, foram então aplicadas as transformadas inversas a este dado. Fez-se o mesmo para o outro atributo. O tamanho do dado sísmico comprimido passa a ser representado pela soma do tamanho dos dois dados comprimidos e é este valor que é utilizado para calcular a taxa de compressão.

Como aqui existem dois dados que variam, para gerar a curva, de taxa de compressão x SNR, fixa-se um atributo e varia-se a quantização do outro atributo. Neste processo, define-se um conjunto de valores que o parâmetro Z pode assumir para o primeiro atributo e outro conjunto de valores que o Z pode assumir para o segundo atributo. Para gerar a curva, faz-se todas as combinações possíveis com os valores de Z dos dois atributos.

Ao gerar a tabela com os valores da taxa de compressão e seus respectivos SNR, haverá muitos valores repetidos. Para gerar os gráficos apresentados neste capítulo foram utilizadas as taxas de compressão que apresentaram o maior valor de SNR, conforme já explicado anteriormente.