# 3
# Uma proposta para compressão de dados sísmicos

Apesar da nova organização proposta melhorar o trabalho com arquivos grandes seu armazenamento e transferência ainda podem ser otimizados com métodos de compressão.

Como estes dados são utilizados em processamentos matemáticos sensíveis que podem apresentar grande diferença entre resultados, para pequenas variações na entrada, é natural que se procure compressão sem perda. Porém a natureza destes dados mostra que este tipo de abordagem não traz resultados significativos.

Os dados sísmicos volumétricos com os quais o trabalho foi realizado são o resultado de duas etapas de processamento. Inicialmente são feitas inúmeras leituras do solo através de sensores. Em seguida os dados dos sensores são cruzados, somados e processados para se obter o valor de cada ponto. No final os valores são armazenados nos formatos típicos de dados sísmicos volumétricos.

A natureza deste processamento juntamente com a precisão com a qual os valores são armazenados aparentam dar ao conjunto um característica aleatória. Vários pontos entre um par de camadas possuem características sísmicas muito parecidas, porém, devido a leves variações do terreno, ruídos e o número de casas decimais registrado na leitura, os valores de suas amostras são semelhantes mas raramente iguais. Em um conjunto de 2 bilhões de amostras, é possível que cada uma delas seja distinta das outras. Como os algoritmos de compressão sem perda dependem de freqüências variáveis para cada amostra esta característica nos leva a crer que a melhor opção é aplicar compressão com perda.

Na literatura encontram-se publicados inúmeros métodos para compressão de dados sísmicos, muitos deles baseados em processamento de sinais e/ou compressão de imagens [7,8]. Apesar disso temos o exemplo do aplicativo de visão v3o2 que emprega uma compressão por agrupamento.

A razão para tal é a maneira que as amostras são utilizadas, a exemplo das aplicações gráficas, deseja-se identificar o valor das amostras associando-as a uma cor. Como o universo de cores utilizadas é em geral menor que o

universo de valores, este último universo é dividido em intervalos e cada intervalo por sua vez é associado a uma cor. Muitas vezes as aplicações limitam-se a utilizar somente tons de cinza resultando em apenas 256 grupos.

Na técnica de agrupamento determina-se um representante para cada grupo e o conjunto é então reescrito substituindo cada amostra pelo representante que mais se aproxima de seu valor original. Este princípio é bom para a compressão de volumes sísmicos porque além de reduzir o número de amostras distintas existente também retira a característica aleatória mencionada.

Neste capítulo discutimos inicialmente duas técnicas de agrupamento; mostramos o agrupamento uniforme e nossa proposta para escolha dos representantes utilizando um método conhecido como K-Medianas. Das seções 3.2.1 à 3.2.4 nos aprofundamos no algoritmo das K-Medianas estudando sua implementação e possíveis otimizações. Na seção 3.3 apresentamos como contornar, com amostragem, a limitação que este algoritmo apresenta com relação ao tamanho da entrada. Uma vez que nossa proposta foi apresentada seguimos comparando-a com o agrupamento uniforme. Apresentamos nossos critérios de avaliação e discutimos as vantagens e desvantagens de cada estratégia. A partir da seção 3.5 discutimos como codificar o dado agrupado a fim de obter o melhor da compressão. Discutimos três métodos para aproveitar a coerência espacial presente em dados sísmicos: a diferença lateral, a diferença segundo a curva de Hilbert e o PPM. Terminamos o capítulo desenvolvendo a implementação de um compressor unindo agrupamento e codificação.



## 3.1. Agrupamento uniforme

A primeira abordagem é fazer um agrupamento linear em relação ao espaço das amostras.

O espaço que vai da menor amostra $a_1$ até a maior $a_n$ é dividido em k partes iguais. Neste caso os representantes de cada grupo podem não fazer parte do conjunto original.

Este critério de agrupamento funciona muito bem para compressão porque, apesar das freqüências serem parecidas para cada valor, este fato não se repete quando tratamos intervalos de valores. Os valores costumam estar concentrados em uma região, em dados sísmicos a distribuição mais comum é gaussiana em torno do zero como visto nos histogramas da Figura 24.

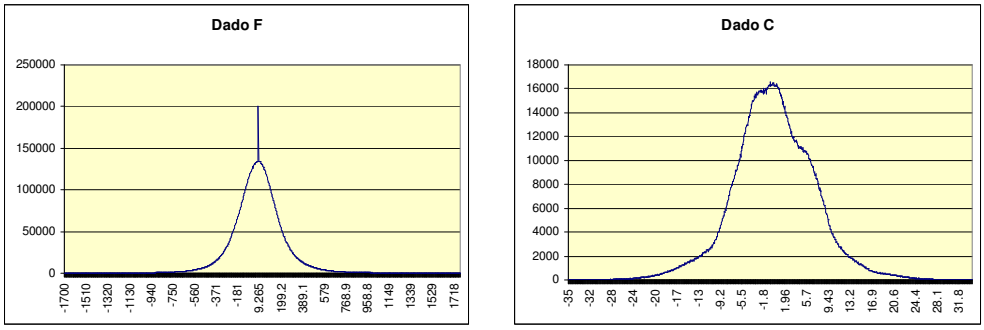



Figura 24: Histograma de dois dados sísmicos.

Esta técnica tem a vantagem de ser simples, mas apresenta alguns problemas. Não há como prever a qualidade do resultado, i.e. o erro introduzido. Os grupos são determinados de acordo com o intervalo de valores e não com sua distribuição. Grupos podem existir sem ter amostras associadas a ele, desperdiçando capacidade de informação. Se a faixa de valores for muito extensa e sua maioria estiver concentrada em poucas regiões, os grupos serão muito largos com poucos deles contendo a maior parte dos dados. Em função destes problemas decidiu-se estudar um método voltado para a minimização do erro médio.

## 3.2. Agrupamento com minimização do erro médio

Uma das maneiras de determinar o agrupamento é escolher os representantes minimizando uma função de custo. Para minimizar o erro médio em um conjunto de valores de uma dimensão temos um algoritmo polinomial baseado em programação dinâmica. Abaixo explicamos a função de custo e como calculamos o conjunto de representantes que minimiza o erro médio.

Dado um grupo de $n$ amostras de $a_1$ até $a_n$ e um único representante $a_r$ o erro médio é dado pelo somatório das distâncias entre cada amostra e o valor do representante dividido pelo número de amostras:

$$\bar{e} = \frac{\sum_{i=1}^{n} \left( |a_i - a_r| \right)}{n} \qquad (5)$$

Para um conjunto de $n$ amostras $a_1,...,a_n$ *com* $a_1 \leq ... \leq a_n$, o representante $a_r$ que minimiza o erro médio é a mediana do grupo.

$$r = \left\lfloor \frac{n}{2} \right\rfloor \quad (6)$$

Dado um conjunto de amostras e um conjunto de representantes, podemos associar cada amostra a um dos representantes formando um agrupamento. Cada grupo é o subconjunto de amostras associadas ao mesmo representante. O erro de cada grupo é dado pela eq. ( 5 ) e o erro do agrupamento é a soma dos erros de todos os grupos. O agrupamento de k grupos ótimo é aquele com menor erro, dentre todos os agrupamentos que utilizam k grupos, podendo existir mais de um.

Sejam $G_1$, $G_2$, ..., $G_k$ os grupos obtidos pelo agrupamento ótimo, é fácil perceber que todas as amostras de cada grupo $G_i$, i=1, ...,k, são contíguas. Com a eq. ( 6 ) vemos que o representante de cada um dos grupos do agrupamento ótimo é a mediana do grupo. Daí nos referenciamos à essa estratégia de agrupamento pelo nome de K-Medianas, ou somente Medianas quando já for implícito o número de grupos.

### 3.2.1. Algoritmo

Para apresentar o algoritmo de K-Medianas vamos primeiro considerar as duas definições a seguir.

Definimos o custo ótimo de agrupar as amostras de $a_i$ até $a_j$ com k grupos como

$$Opt(i, j, k) \quad (7)$$

O custo de montar um único grupo com amostras de $a_i$ até $a_j$ é dado por

$$C(i, j) = \frac{\sum_{x=i}^{j} \left( \left| a_x - a_{\left\lfloor \frac{i+j}{2} \right\rfloor} \right| \right)}{j - i + 1} \quad (8)$$

Confirme dito anteriormente trata-se de um algoritmo de programação dinâmica, ou seja, a solução desejada é encontrada a partir de outras soluções previamente calculadas. Neste caso consideramos que conhecemos a melhor maneira de agrupar x amostras com k-1 representantes, para x variando de 1 a n

e assim podemos descobrir a melhor maneira de agrupar n amostras com k representantes.

O novo representante, i.e. o k-ésimo representante, agrupará as i últimas amostras e os demais representantes agruparão, da melhor maneira possível, as n-i primeiras amostras utilizando k-1 representantes. Resta descobrir o valor de i que minimiza o custo total do agrupamento.

A equação de recorrência do algoritmo é então definida

$$Opt(1,n,k) = \underset{i=n-1,\ldots,1}{MIN}\{Opt(1,i,k-1) + C(i+1,n)\} \quad (9)$$

Os casos base são:

Para um único grupo a solução ótima é igual ao custo.

$$Opt(i,j,1) = C(i,j) \quad (10)$$

Para uma amostra o custo é zero.

$$C(i,i) = 0 \quad (11)$$

### 3.2.2. Implementação não recursiva

A primeira implementação realizada usa a memória para guardar os valores das etapas anteriores da recorrência e os grupos escolhidos. São utilizadas duas matrizes de dimensões n *x* k e a execução ocorre em três iterações encapsuladas. Esta implementação é uma tradução direta da eq. ( 9 ).

Pseudo-código

```
otimos [n][k] = Matriz nxk de custos ótimos.
ultimos[n][k] = Matriz nxk com  os últimos de cada grupo.

para i crescendo de 1 a n
 |
 | otimos[i][1] = custo(1, i) ;

para i de 1 a k
 |
 |  otimos [1][i] = 0 ;

para l crescendo de 1 até k
 |
 | para i crescendo de 2 até n
 |  |
 |  | min   = infinito
 |  | min_m = -1 ;
 |  |
 |  | para m decrescendo de i até 2
 |  |  |
 |  |  | val = otimos[m-1][l-1] + custo( m, i )
 |  |  |
 |  |  | se val < min
 |  |  |  |
 |  |  |  | min_m = m-1 ;
 |  |  |  | min = val ;
 |  |
 |  | ultimos[i][l] = min_m
 |  | otimos[i][l]  = min
```

Para recuperar o valor dos representantes caminhamos na matriz *ultimos* montando os grupos.

Pseudo-código

```
representantes[k]

   ult_do_grupo = n
   ult_do_grupo_anterior = ultimos[n][k]
   for r = k até 1
   {
      ult_do_grupo_anterior = ultimos[ultima_do_grupo][k]

      representantes[r] =
                ( ult_do_grupo_anterior+1 + ult_do_grupo )/2

      ult_do_grupo = ult_do_grupo_anterior
   }
```

Analisando a complexidade computacional da implementação temos que a ordem de tempo é $O(kn^2)$ e o gasto com memória é O(kn), o que é consideravelmente alto. Para uma execução que visa encontrar 512 grupos para 64 mil amostras são necessários 125Mb de memória para cada uma das matrizes.

### 3.2.3.
### Implementação recursiva

Embora a solução anterior possa ser facilmente modificada para que se calcule o custo ótimo com apenas dois vetores de tamanho n e, assim, eliminar a matriz *custo*, ainda precisaríamos da matriz *ultimos* para recuperar os representantes. Para eliminar este problema adotamos a estratégia recursiva proposta por Hirschberg [4]. Esta estratégia é baseada na seguinte identidade:

$$Opt(1,n,k) = \underset{i=0}{\overset{n}{MIN}} \left\{ Opt\left(1,i,\left\lfloor \frac{k}{2} \right\rfloor\right) + Opt\left(i+1,n,\left\lceil \frac{k}{2} \right\rceil\right) \right\} \qquad (12)$$

Onde *Opt*( *i,j, k* ) = 0 se *j*≤*i*

Nesta implementação recursiva a técnica de dividir e conquistar é aplicada. Primeiramente calculamos $Opt(1,i,\lfloor k/2 \rfloor)$ e $Opt(i,n,\lceil k/2 \rceil)$ para *i*=1,...,*n*. Note que estes valores podem ser obtidos em $O(kn^2)$ e com gasto de espaço linear usando a estratégia proposta na seção anterior. A partir de então utilizamos a eq. ( 12 ) e obtemos i* tal que $Opt(1,i*,\lfloor k/2 \rfloor)$ + $Opt(i*+1,n,\lceil k/2 \rceil)$ = $Opt(1,n,k)$. Este índice pode ser obtido em $O(n)$. Então i* é armazenado como o último índice do $\lfloor k/2 \rfloor$-ésimo grupo na solução ótima. Finalmente o procedimento é chamado recursivamente para calcular os delimitadores, i.e. a última amostra de cada grupo, do agrupamento ótimo para as i* primeiras amostras em $\lfloor k/2 \rfloor$ grupos e os delimitadores do agrupamento ótimo das i*+1, ..., n amostras em $\lceil k/2 \rceil$ grupos.

Esta implementação tem custo de tempo de mesma ordem que a solução anterior. Os resultado dos testes de execução podem ser vistos nos gráficos das Figura 25 e Figura 26 a seguir.

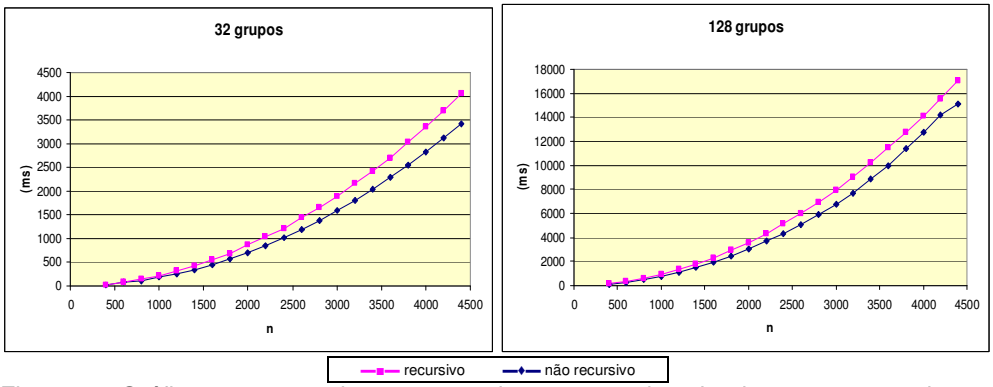



Figura 25: Gráficos comparando os tempos de execução dos algoritmos não recursivo e recursivo em função do tamanho do conjunto de amostras com um número fixo de representantes. A esquerda estão os resultados do teste para 32 representantes e a direita para 128.

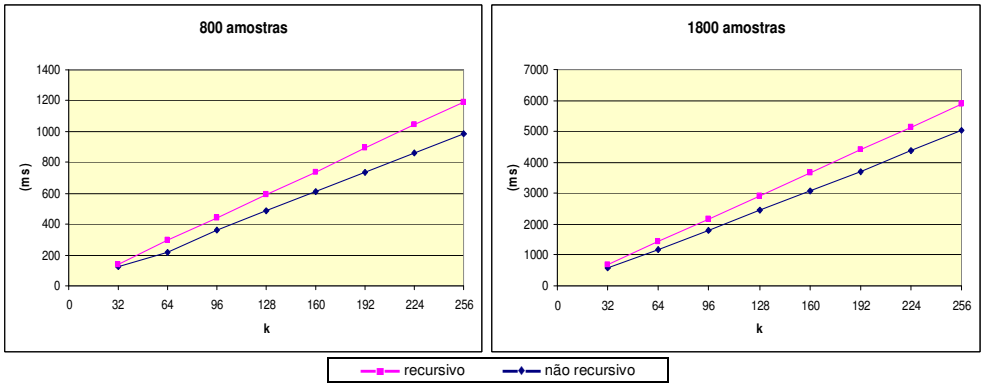



Figura 26: Gráficos comparando os tempos de execução dos algoritmos não recursivo e recursivo em função do número de grupos para um tamanho fixo do conjunto de amostras. A esquerda estão os resultados do teste para 800 amostras e a direita para 1800.

Os testes realizados mostram que a diferença entre as constantes escondidas na notação *O()* dos dois algoritmos é pequena. Embora o algoritmo recursivo tenha demonstrado ser um pouco mais lento o tempo total perdido ainda é baixo e o ganho com memória obtido justifica o uso desta implementação. Com o algoritmo não recursivo a limitação de memória não nos permitiu encontrar o agrupamento ótimo para mais que 64 mil amostras quando o número de grupos valia 512. Com o algoritmo recursivo chegamos a testar, com êxito, uma execução com 200 mil amostras e 512 grupos.

### 3.2.4.
### Otimizando o tempo de execução

Em cada etapa do algoritmo em que queremos acrescentar um grupo novo, procuramos descobrir quais últimas amostras estarão neste novo grupo. São consideradas então todas as possibilidades. Existe, porém, uma forma de cortar este espaço de busca.

Vamos, agora, analisar a recorrência da eq. ( 9 ) tentando entender como a solução muda ao acrescentarmos uma nova amostra para um número fixo de grupos.

Conhecidos os representantes necessários para agrupar, de maneira ótima, j amostras, $a_1,...,a_j$, ordenadas em k grupos, introduz-se uma nova amostra $a_{j+1} \geq a_j$. Conforme já foi mencionado anteriormente, os grupos ótimos contem amostras contíguas, então $a_{j+1}$ deve estar contida dentro do último grupo. Além disso vamos mostrar que, com esta nova inclusão, sempre conseguimos encontrar uma solução ótima deslocando a posição do último representante para uma amostra de maior valor, não sendo necessário comparar com soluções que o desloquem para amostras menores. O início do último grupo deve ser portanto deslocado apenas no mesmo sentido. Um exemplo ilustrativo deste processo pode ser visto na Figura 27. O Lema 1, a seguir, define esta propriedade.

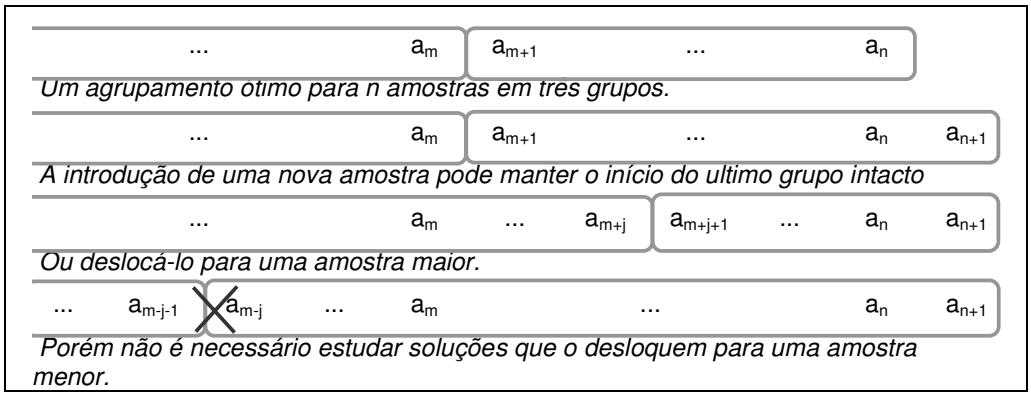



Figura 27: Exemplo do movimento dos grupos com a introdução de uma nova amostra

Lema 1:

> Existe solução ótima para agrupar n+1 amostras em k grupos onde início do último grupo não é menor que o início do último grupo da solução ótima para agrupar n amostras em k grupos

Prova:

Sejam
*Opt( 1, n, K ) = Opt( 1, m-1, K-1 ) + C( m, n )*
e
*Opt( 1, n+1, K ) = Opt( 1, m'-1, K-1 ) + C( m', n+1 )*
*Onde m'< m*

Definimos o custo dessa solução como:
*Custo( Opt( 1, n+1, K) )*
*= Custo( Opt( 1, m'-1, K-1 )) + C( m', n+1 )*
*= Custo( Opt( 1, m'-1, K-1 )) + C( m', n ) + Dist( $a_{(m'+n)/2}$ , $a_{j+1}$ ) ≥*

Como
*custo( Opt( 1, n, K ) ) = Custo( Opt( 1, m-1, K-1 )) + C( m, n )*
então
*Custo( Opt( 1, m'-1, K-1 )) + C( m', n ) = custo( Opt( 1, n, K ) ) ≥*
*Custo( Opt( 1, m-1, K-1 )) + C( m, n ) = custo( Opt( 1, n, K ) )*

Como
*m' < m* e *$a_i \le a_{i+1}$ para i = 1..n-1*
então
*Dist( $a_{(m'+n)/2}$ , $a_{j+1}$ ) ≥ Dist( $a_{(m+n)/2}$ , $a_{j+1}$ )*

portanto
*Custo( Opt( 1, m-1, K-1 )) + C( m, n ) + Dist( $a_{(m+n)/2}$ , $a_{j+1}$ ) ≥*
*Custo( Opt( 1, m'-1, K-1 )) + C( m', n ) + Dist( $a_{(m'+n)/2}$ , $a_{j+1}$ )*

Então existe solução ótima para agrupar n+1 amostras em K grupos onde início do ultimo grupo não é menor que o início do ultimo grupo da solução ótima para agrupar n amostras em K grupos

Esta propriedade limita o conjunto de amostras que podem fazer parte do novo grupo introduzido a cada etapa da equação de recorrência original. Ganha-se com redução no número de comparações e conseqüentemente redução no tempo total de execução.

Seja $r_{ijk}$ o índice da primeira amostra do k-ésimo grupo do agrupamento ótimo das amostras no intervalo $[a_i,a_j]$ em k grupos, então:

$$Opt(1,n,k) = \underset{i=n-1,\ldots,r_{1,n-1,k}}{MIN} \{Opt(1,i,k-1) + C(i+1,n)\} \qquad (13)$$

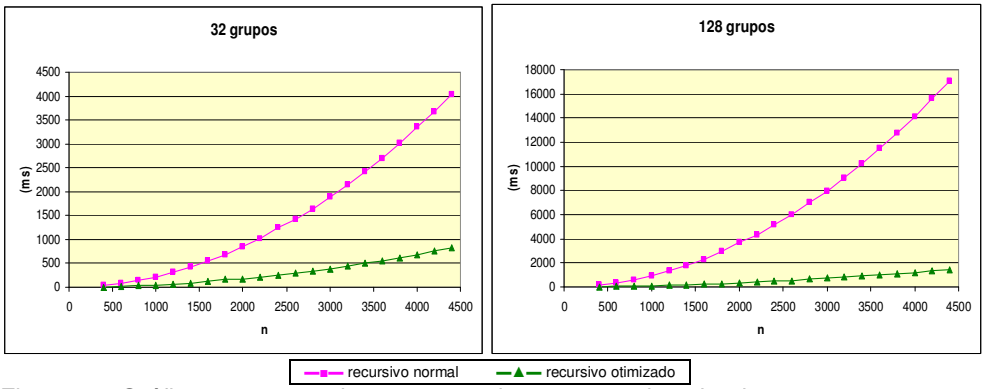



Figura 28: Gráficos comparando os tempos de execução dos algoritmos com e sem o corte no espaço de busca em função do tamanho do conjunto de amostras com um número fixo de representantes.

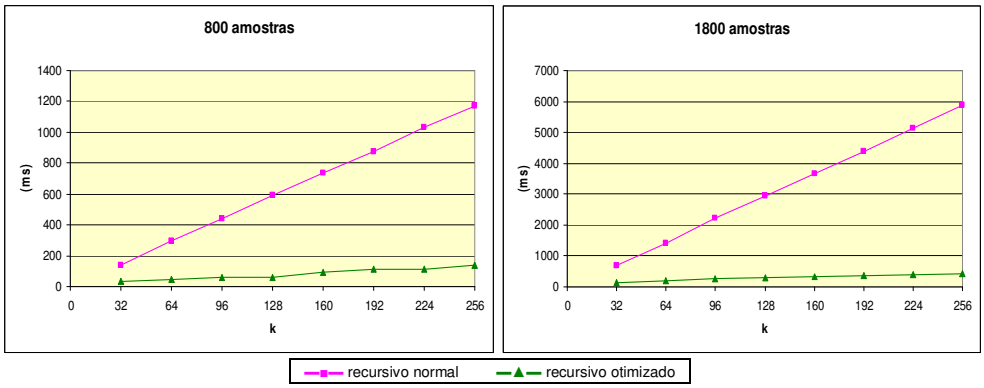



Figura 29: Gráficos comparando os tempos de execução dos algoritmos com e sem o corte no espaço de busca em função do número de grupos para um tamanho fixo do conjunto de amostras.

De acordo com os testes mostrados nos gráficos acima a razão do ganho cresce tanto com o número de amostras quanto com o número de grupos. Com 1.800 amostra e 256 grupos tempo da versão otimizada chegou a ser doze vezes menor, como vemos na Figura 29. Embora no pior caso esta estratégia otimizada não apresente ganho, nos casos testados a variação é bem significativa.

## 3.3. Amostragem

As ordens de tempo e memória do algoritmo tornam impraticável sua aplicação no volume total de dados. Conforme mencionado, os arquivos com os quais trabalhamos podem chegar a dezenas de bilhões de amostras. Esta

quantidade de dados não pode ser mantida em memória. Entretanto, uma solução com uso de disco prejudicaria consideravelmente o tempo de execução.

Estudamos, então, uma proposta de aplicar o algoritmo sobre uma amostragem aleatória dos dados. Os representante são determinados para esta sub-seleção e posteriormente utilizados para classificar o arquivo todo. A solução final deixa de ser ótima, porém, como a amostragem tende a ser de boa qualidade devido ao fato dos dados sísmico serem comportados, a diferença não será considerável.

Para montar o sub-conjunto de amostras o dado inteiro é lido. Cada amostra considerada é incluída na seleção com probabilidade igual a razão entre o tamanho da amostragem e o total de amostras no arquivo. A implementação desta probabilidade foi feita utilizando um método pseudo-aleatório fornecido pela linguagem de programação utilizada.

O gráfico a seguir mostra o erro médio da quantização no volume todo em função do tamanho da sub-amostragem utilizada para encontrar os grupos. O mesmo tipo de teste foi realizado para quantizações com 128, 256 e 512 grupos em dois volumes diferentes. Cada ponto das curvas mostra, no eixo vertical, o erro absoluto médio da quantização no arquivo todo, utilizando os representantes encontrados a partir de uma sub-amostragem cujo tamanho encontra-se no eixo horizontal.

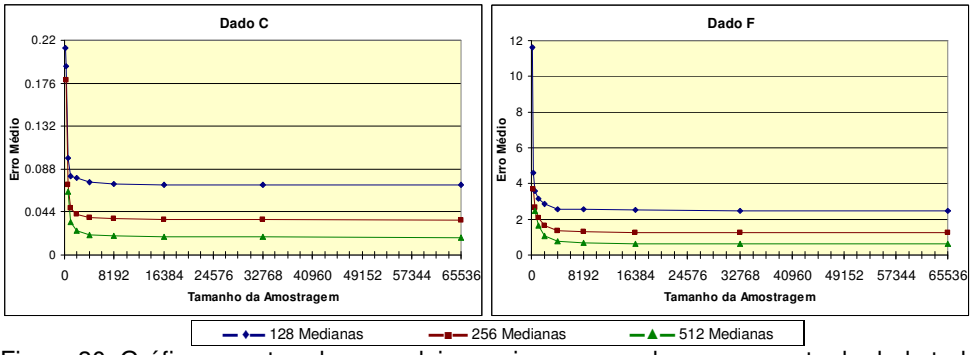



Figura 30: Gráficos mostrando, para dois arquivos, o erro do agrupamento do dado todo em função do tamanho da amostragem utilizada para determinar os representantes.

O gráfico da Figura 30 mostra claramente o decaimento do erro da quantização em função do tamanho da amostragem. Este decaimento, como pode ser observado, não é linear. Em um determinado momento as curvas de nossos testes aparentam atingir um ponto de estabilidade demonstrando que a partir deste ponto o ganho com uma maior amostragem é desprezível. Este

comportamento é esperado para um conjunto de dados comportado e amostragem aleatória.

Uma forma de mostrar qual é realmente o limite do erro da quantização sugerido nos gráficos anteriores é comparar o custo de agrupar somente a amostragem com os representantes escolhidos pelo algoritmo com o custo de agrupar o arquivo todo com os mesmos representantes.

Os representantes escolhidos geram o agrupamento ótimo para as amostras da sub-seleção. Ao utilizar estes mesmos representantes para agrupar um dado maior que a amostragem não se espera que o erro médio seja menor. Isto poderia ser verdade em alguns casos se, por exemplo, toda amostra do dado que não pertencesse a amostragem tivesse exatamente o valor de um representante. Cada nova amostra somaria erro zero e o erro médio, no final, seria menor. Porém, como estamos tratando de dados bem comportados e de uma amostragem aleatória pode-se desprezar a possibilidade desta ocorrência. O erro médio do uso dos representantes em um conjunto maior que o utilizado para determiná-los tende a ser sempre maior que o erro médio neste último conjunto. Os gráficos da Figura 31 comparam a evolução deste erros em função do tamanho da sub-amostragem. No começo, quando o número de amostras selecionadas é igual ao número de grupos os representantes escolhidos ficam sendo justamente essas amostras. O erro de agrupar a seleção é zero, mas ao usar estes representantes para agrupar o arquivo o erro resultante é bem alto. A medida que a amostragem aumenta fica mais difícil escolher os representantes e cresce o erro de agrupar somente este sub-conjunto, porém decresce o erro de usar estes representantes no arquivo todo. A partir de um determinado ponto estes dois erros são muito próximos sugerindo que a amostragem já é uma boa representação do dado.

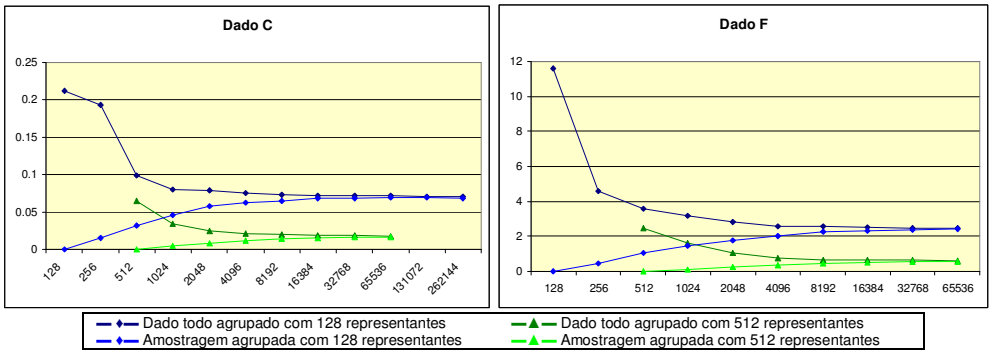



Figura 31: Os gráficos comparam o erro do agrupamento do dado todo com o erro de agrupamento da sub-amostragem utilizando o mesmo grupo de representantes em função do tamanho da amostragem.

## 3.4.
## Comparação do agrupamento uniforme com K-medianas

Para testar a qualidade do resultado obtido com a técnica de agrupamento das K-Medianas fizemos uma comparação com o agrupamento por intervalos uniformes. Os critérios utilizados foram o erro médio e a capacidade de compressão medida com a entropia. A entropia de codificação é um bom estimador da redução que se pode atingir através de métodos como Huffman [7,11] e compressão aritmética [7,11,12,13]. Mais precisamente, a entropia consegue indicar o tamanho médio, em bits, que cada símbolo ocupará ao comprimir um conjunto de dados.

O dado original, em ponto flutuante, é guardado com uma codificação de 32 bits por valor. Após o agrupamento o dado pode ser reduzido utilizando uma codificação de tamanho fixo em que cada símbolo ocupa $\lceil \log_2(k) \rceil$ bits, onde *k* é o número de grupos. Neste caso compararíamos as duas técnicas apenas com o erro. Como queremos, entretanto, atingir o máximo de compressão possível consideramos a aplicação de um método tradicional de compressão no dado agrupado e utilizamos a entropia para entender como o método é afetado pelas técnicas de agrupamento em questão.

O tamanho do código associado a cada símbolo, após a compressão, depende de sua freqüência. A entropia de um conjunto de dados qualquer com k símbolos distintos é dada por

$$Ent = -\sum_{i=1}^{k} \log_2(p_i) \cdot p_i \qquad (14)$$

onde *k* é o número de símbolos distintos existentes no conjunto e $p_i$ a probabilidade do símbolo *i* ocorrer no conjunto equivalendo à razão entre o número de suas ocorrências $o_i$ e o total de amostras *n*,

$$p_i = \frac{o_i}{n} \qquad (15)$$

Para ilustrar a medição do potencial de compressão com entropia, vamos supor que queremos comprimir o primeiro parágrafo deste texto, onde por simplicidade, retiramos os acentos e as cedilhas resultando em:

> os dados sismicos guardam informacoes colhidas sobre o subsolo de uma regiao e sao utilizados pelos geofisicos para analisar e estudar tal regiao de terreno. os objetivos variam desde estudo sobre ocorrencias de atividades sismicas como vulcoes e terremotos ate a identificacao de possiveis pocos de combustiveis fosseis.

O alfabeto utilizado tem vinte e cinco símbolos: as letras do alfabeto da língua portuguesa, o espaço e o ponto, e o texto tem 321 caracteres no total. Uma codificação de tamanho fixo necessita de $\lceil \log_2(25) \rceil = 5$ bits por símbolo. O texto total deve ter, então, 1605 bits de comprimento.

Agora vejamos como estes símbolos estão distribuídos no texto. A tabela abaixo mostra o número de ocorrências de cada símbolo no parágrafo.

| Letra | a | b | c | d | e | f | g | h | i | j | l | M | n |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Ocorr. | 27 | 5 | 13 | 17 | 32 | 4 | 4 | 1 | 26 | 1 | 7 | 9 | 5 |

| Letra | o | p | q | r | s | T | u | v | x | z | espaço | . | total |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Ocorr. | 36 | 4 | 0 | 16 | 38 | 12 | 8 | 6 | 0 | 1 | 47 | 2 | 321 |

Tabela 9: Número de ocorrências dos de cada símbolo do texto do exemplo.

A probabilidade de ocorrência da letra *c*, por exemplo, no texto é 13 em 321 enquanto que para a letra *s* esta probabilidade é 38 em 321. Como *q* e *x* não ocorrem sua probabilidade é zero e o termo $log_2(p_i)*p_i$ equivalentes a estes caracteres podem ser eliminados da computação da entropia do conjunto.

A entropia do exemplo acima, então, é calculada com:

$$-(\log_2(p(a))\cdot p(a)+\log_2(p(b))\cdot p(b)+\cdots+\log_2(p(z))\cdot p(z)+\log_2(p(\ ))\cdot p(\ )+\log_2(p(\bullet))\cdot p(\bullet))=$$

$$-\left(\log_2\left(\frac{27}{321}\right)\cdot\frac{27}{321}+\log_2\left(\frac{5}{321}\right)\cdot\frac{5}{321}+\cdots+\log_2\left(\frac{1}{321}\right)\cdot\frac{1}{321}+\log_2\left(\frac{47}{321}\right)\cdot\frac{47}{321}+\log_2\left(\frac{2}{321}\right)\cdot\frac{2}{321}\right)=$$

$$-(-0.3004-0.0936-\cdots-0.0259-0.4059-0.0457)=3.9067$$

Isso quer dizer que o texto pode ser escrito com uma média de 3.9067 bits por caractere, deixando o texto total com comprimento de 1255 bits em vez dos 1605 necessários para um código de tamanho fixo.

A entropia não diz como cada símbolo deve ser codificado, mas informa qual grau de compressão pode-se atingir a partir da probabilidade de cada símbolo. Neste trabalho propomos que um dos métodos clássicos de codificação seja aplicado após o agrupamento. Para saber o quanto, em capacidade de compressão, os agrupamentos oferecem, vemos cada representante como um símbolo e calculamos a entropia do dado agrupado.

Com o número *k* de grupos fixo, a comparação entre as duas técnicas sempre parece favorável para as K-Medianas, quando o critério é erro médio, porém a técnica de k intervalos uniformes apresenta melhor capacidade de compressão. Estas comparações podem ser vistas na Tabela 10.

| Dado | k | Erro médio | | Entropia | |
|---|---|---|---|---|---|
| | | Intervalos uniformes | Medianas | Intervalos uniformes | Medianas |
| F | 256 | 3,39040 | 1,25105 | 5,980 | 7,708 |
| | 512 | 1,70108 | 0,63531 | 6,983 | 8,715 |
| | 1024 | 0,84813 | 0,32354 | 7,983 | 9,717 |
| R | 256 | 0,00281 | 0,00065 | 5,143 | 7,672 |
| | 512 | 0,00140 | 0,00034 | 6,145 | 8,664 |
| | 1024 | 0,00070 | 0,00018 | 7,146 | 9,689 |

Tabela 10: Comparação entre agrupamento por intervalos uniformes e medianas utilizando os critérios de erro médio e entropia.

Estes dois critérios têm uma característica compensatória. A tentativa de melhorar um deles tende a prejudicar o outro. Quando comparamos as duas técnicas, sem considerar o número de grupos utilizados, vemos claramente a existência da relação entropia *x* erro. A Tabela 11 mostra que, se aumentarmos suficientemente o número de grupos do agrupamento por intervalos uniformes, atingimos o mesmo erro médio obtido pelo agrupamento das K-Medianas com menos grupos. Neste ponto a entropia do dado também atinge nível semelhante.

| Dado | Medianas | | | Intervalos Uniformes | | |
|---|---|---|---|---|---|---|
| | K | Erro Médio | Entropia | K | Erro Médio | Entropia |
| F | 128 | 2,4752 | 6,7363 | 320 | 2,7206 | 6,3038 |
| | 256 | 1,2546 | 7,7269 | 640 | 1,3544 | 7,3055 |
| | 512 | 0,6330 | 8,7185 | 1408 | 0,6182 | 8,4434 |
| R | 128 | 0,071623 | 6,835778 | 512 | 0,068104 | 6,692945 |
| | 512 | 0,018295 | 8,80396 | 896 | 0,019407 | 8,50413 |
| | 896 | 0,010607 | 9,62424 | 1600 | 0,010855 | 9,341199 |

Tabela 11: Comparação entre o número de grupos que cada técnica precisa para atingir níveis semelhantes de erro médio e entropia.

A escolha da técnica utilizada será determinada de acordo com o critério considerado mais importante. Em algumas aplicações o agrupamento não é utilizado apenas para compressão. No caso de aplicações gráficas, por exemplo,

cada grupo é associado a uma cor para o desenho do terreno. Em aplicações de inteligência e classificação automática são utilizados representantes pois seria impraticável a execução dos algoritmos com os valores originais. Nestes últimos exemplos, o número de grupos é limitado, e pode ser considerado o fator de decisão. O método mais recomendado então é o agrupamento por medianas que induz a um erro menor.

Nos casos gerais, a determinação de erro máximo ou compressão mínima desejados serão responsáveis por determinar o número de grupos. Como a relação entre estes critérios não varia muito com a técnica utilizada, ambas trarão resultados parecidos.

## 3.5. Transformando o dado através da diferença lateral

Os métodos de agrupamento apresentados atuam sobre o conteúdo armazenado mas não se aproveitam da maneira como estes são organizados ou da sua natureza. Por si estes métodos já seriam suficientes para comprimir o dado, porém como conhecemos bem sua natureza estudamos novas técnicas que visam ganhar ainda mais na compressão, sem introduzir novas perdas de informação, em função de particularidades do dado sísmico.

Os volumes de dados sísmicos representam uma região contínua do terreno. Isso significa que há coerência entre os valores de amostras próximas. Na direção vertical o dado tem característica ondulatória, variando rapidamente a intensidade e o sentido. Nas direções laterais as variações ocorrem de acordo com inclinação do terreno com diferenças altas apenas nos pontos de falhas, canais ou outras estruturas geológicas mais raras. A Figura 32 ilustra isso.

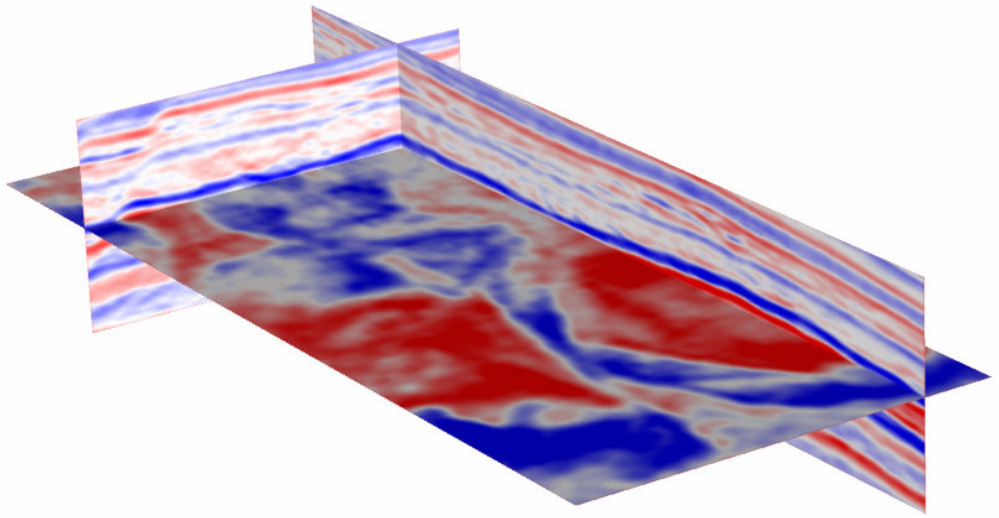


Figura 32: Coerência horizontal maior que coerência vertical .

Se os valores são parecidos, então a diferença entre eles é baixa. Calculando a diferença entre um ponto e outro ao seu lado transformamos os dados de forma a aumentar a freqüência dos índices de baixo valor e conseqüentemente reduzir a entropia. O exemplo da Figura 33 ilustra esse procedimento. A operação de diferença é reversível e não acrescenta mais perda de precisão à compressão.

| A | B | C |
|---|---|---|
| 1 | 1 | -1 |
| -2 | 1 | 0 |
| 5 | 4 | 5 |
| 6 | 7 | 6 |
| 9 | 9 | 10 |
| 5 | 6 | 9 |
| 1 | 0 | 7 |
| -8 | 1 | 2 |
| -12 | -9 | -4 |
| -9 | -11 | -8 |

| A | B-A | C-B |
|---|---|---|
| 1 | 0 | -2 |
| -2 | 3 | -1 |
| 5 | -1 | 1 |
| 6 | 1 | -1 |
| 9 | 0 | 1 |
| 5 | 1 | 3 |
| 1 | -1 | 7 |
| -8 | 9 | 1 |
| -12 | 3 | 5 |
| -9 | -2 | 3 |

Figura 33: Sejam A, B e C os três primeiros traços de um dado sísmicos após a aplicação de um agrupamento. Enquanto o dado original (esquerda) guarda os grupos o dado de diferenças (direita) guarda o primeiro traço e a variação para os seguintes.

A operação de diferença não é aplicada diretamente no arquivo original, e sim no arquivo quantizado, para que não haja acúmulo de erro na reconstrução do dado original. A estratégia pode ser aplicada em qualquer dado independentemente da técnica de quantização utilizada. Em testes realizados a estratégia apresentou ganho significativo tanto para quantização uniforme

quanto de medianas. Os gráficos da Figura 34 mostram os histogramas dos valores no dado original e no dado após o cálculo da diferença.

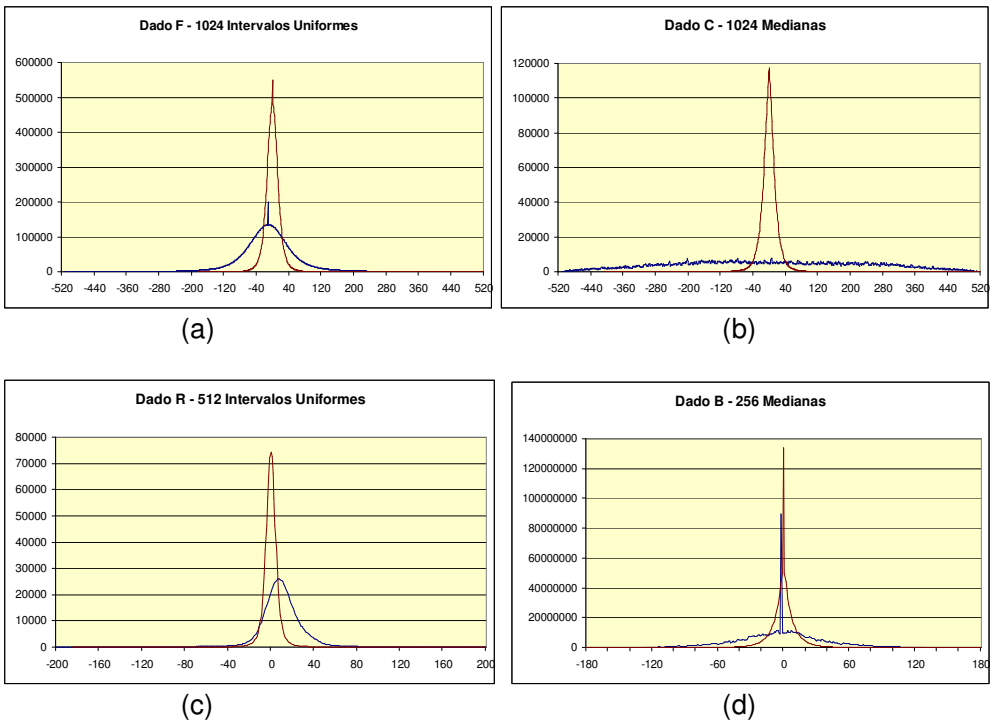



Figura 34: Em cada um dos quadros vemos, em azul o histograma do resultado do agrupamento e, em vermelho, o histograma do mesmo dado após a diferença lateral. Da esquerda para a direita, de cima para baixo: (a) Dado F agrupado em 1024 intervalos uniformes, (b) Dado C agrupado em 1024 medianas, (c) Dado R agrupado em 512 intervalos uniformes, (d) Dado B agrupado em 256 medianas

Inicialmente parece que a diferença seria prejudicial para a compressão porque o universo de valores dobra, porém para compressão a extensão deste universo não é tão importante quanto a distribuição das amostras dentro dele. Para determinar a melhora no potencial de compressão calculamos a entropia dos dados antes e após a diferença. Comparamos o efeito da diferença para agrupamentos uniforme e de medianas para vários valores de k.

| Dado | Tipo de agrupamento | Número de grupos | Entropia do dado agrupado | Entropia do dado com diferença lateral | Razão |
|---|---|---|---|---|---|
| R | medianas | 1024 | 9,6886 | 8,2306 | 84,95% |
| | | 512 | 8,6640 | 7,1948 | 83,04% |
| | | 128 | 6,6468 | 5,1769 | 77,88% |
| | uniforme | 1024 | 7,1458 | 5,6489 | 79,05% |
| | | 512 | 6,1450 | 4,6510 | 75,69% |
| | | 128 | 4,1412 | 2,7061 | 65,34% |
| C | medianas | 1024 | 9,8092 | 6,2033 | 63,24% |
| | | 512 | 8,8038 | 5,1909 | 58,96% |
| | | 128 | 6,8209 | 3,2331 | 47,40% |
| | uniforme | 1024 | 8,6969 | 5,0364 | 57,91% |
| | | 512 | 7,6957 | 4,0422 | 52,53% |
| | | 128 | 5,6877 | 2,1605 | 37,98% |
| F | medianas | 1024 | 9,7169 | 7,9224 | 81,53% |
| | | 512 | 8,7155 | 6,9163 | 79,36% |
| | | 128 | 6,7252 | 4,9155 | 73,09% |
| | uniforme | 1024 | 7,9839 | 6,0506 | 75,78% |
| | | 512 | 6,9832 | 5,0509 | 72,33% |
| | | 128 | 4,9764 | 3,0741 | 61,77% |

Tabela 12: Entropia de 3 dados após a aplicação de diferentes agrupamento e após aplicação da diferença lateral no resultado do agrupamento. A última coluna mostra a razão entre os tamanhos estimados para os dados se comprimidos após a operação de diferença ou somente com o agrupamento.

Em qualquer dos casos testados a diferença lateral gera um dado que pode ser melhor comprimido que o dado resultante do agrupamento original. No melhor caso, para o dado C com agrupamento em 128 intervalos uniformes, o novo dado pode chegar a ser reduzido para 38% do tamanho que atingiria comprimindo. E mesmo no pior caso, mostrado na primeira linha, há uma redução de 15%, já sendo bastante significativa.

A estratégia de operar com diferença parece ser uma boa escolha para atingir melhores taxas de compressão. É interessante lembrar que as razões apresentadas na última coluna da Tabela 12 mostram o ganho sobre o agrupamento que por sua vez já apresenta uma redução sobre o dado original em ponto flutuante.

## 3.6. Curva de Hilbert

Na estratégia de diferença lateral uma direção é determinada e a diferença é sempre feita nesta direção. Como não há, para dados sísmicos, como

determinar ou afirmar se a coerência em uma direção é maior que outra, a qualidade dos resultados varia caso a caso.

Observando alguns dados é natural acreditar que se permanecermos, o máximo possível, em uma região próxima antes de passar para outra conseguiremos obter valores parecidos e, conseqüentemente, diferenças baixas. Existem maneiras conhecidas de se caminhar por uma curva e preencher toda uma área quadrada, as curvas de Hilbert, de Peano e de Sierspinski são todas exemplos válidos [7]. Para comparar a estratégia da diferença lateral com outra estratégia este trabalho elaborou testes com a curva de Hilbert.

A curva de Hilbert é uma curva que preenche um espaço. Dada uma grade quadrada cujo lado é potência de 2 pode-se, começando em um vértice, percorrer todo o espaço sem andar mais de três vezes consecutivas na mesma direção. As curvas são classificadas por sua ordem $o = log_2(l)$ onde $l$ é o tamanho do lado do quadrado. Todas as curvas de Hilbert são composições de diferentes rotações da curva da ordem anterior até a curva básica de primeira ordem como mostra a Figura 35.

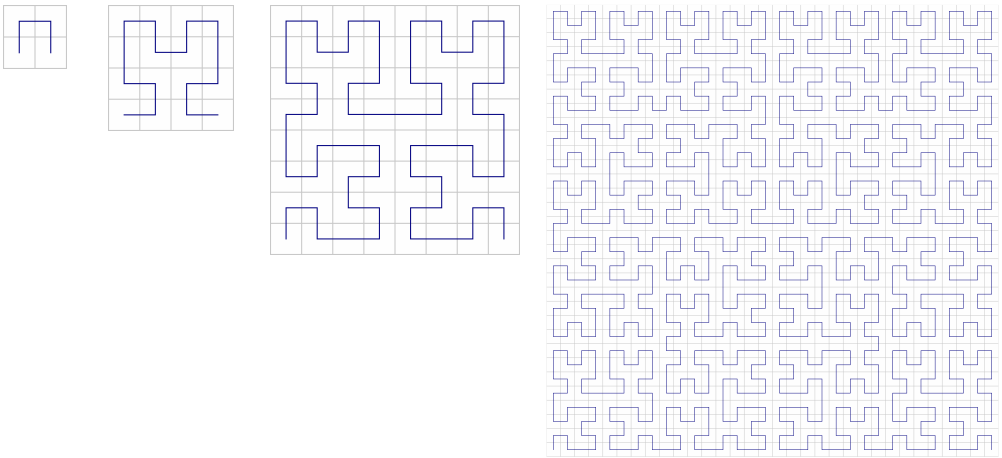
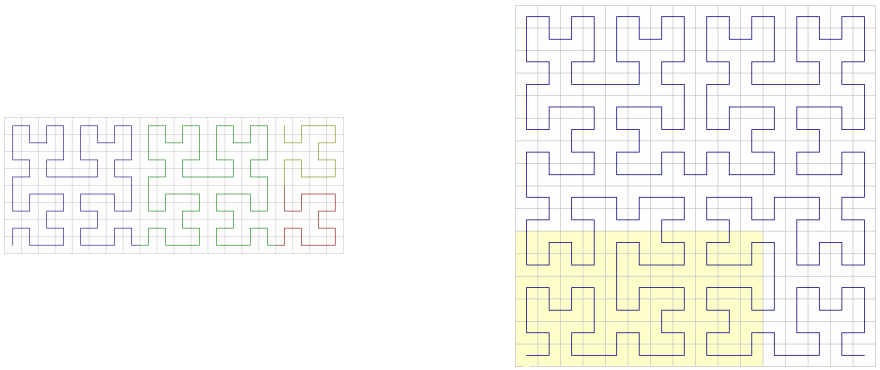


Figura 35: Da esquerda para a direita temos as curvas de primeira, segunda, terceira e quinta ordem. Dividindo uma curva em quatro os quadrantes superiores possuem cópias exatas da curva da ordem anterior enquanto que os quadrantes inferiores possuem rotações de 90º e 270º da mesma.

A idéia é caminhar, seguindo a curva de Hilbert, nos planos *z=cte* do dado e ir guardando a diferença entre o valor da célula atual pra célula anterior. Espera-se, assim, resultados em média melhores ou iguais a diferença unidirecional.

O problema da curva de Hilbert é o formato da área preenchida, que deve ser sempre um quadrado de lado potência de 2. Para este problema existe uma gama de soluções propostas na literatura. A área pode ser preenchida com a união de curvas de várias ordens (Figura 36.a), neste caso a restrição de não caminhar mais que três vezes consecutivas na mesma direção poderá ser violada se a dimensão de um dos lados for ímpar. Outra solução é utilizar uma curva cuja ordem contenha toda a área do dado e fixar um valor para os pontos que extrapolam a dimensão do dado sísmico (Figura 36.b). Para nossos testes consideramos apenas dados com lado potência de 2.



(a) (b)

Figura 36: (a) Área coberta por duas curvas de 3ª ordem e duas curvas de 2ª
(b) Área coberta pela menor curva que a contém.

Dos dois dados utilizados um deles foi cortado para que seu tamanho fosse compatível com o algoritmo. A Tabela 13 mostra as entropias originais, com diferença Lateral e de Hilbert e a comparação entre elas. De maneira geral não houve ganho com esta estratégia.

| Dado | Tipo do agrupa-mento | k | Entropia do dado agrupado | Entropia do dado com diferença lateral | Entropia do dado com diferença de Hilbert | Razão com diferença lateral | Razão com diferença de Hilbert |
|---|---|---|---|---|---|---|---|
| C com corte | medianas | 1024 | 8,5630 | 6,3265 | 6,8228 | 73,88% | 79,68% |
| | | 512 | 7,5618 | 5,3126 | 5,8080 | 70,26% | 76,81% |
| | | 128 | 5,5539 | 3,3495 | 3,8349 | 60,31% | 69,05% |
| | uniforme | 1024 | 9,7391 | 6,3265 | 6,8228 | 64,96% | 70,06% |
| | | 512 | 8,7318 | 5,3126 | 5,8080 | 60,84% | 66,52% |
| | | 128 | 6,7497 | 3,3495 | 3,8349 | 49,62% | 56,82% |
| F | medianas | 1024 | 9,7169 | 7,9224 | 8,1499 | 81,53% | 83,87% |
| | | 512 | 7,9839 | 6,0506 | 6,2851 | 75,78% | 78,72% |
| | | 128 | 8,7155 | 6,9163 | 7,1437 | 79,36% | 81,97% |
| | uniforme | 1024 | 6,9832 | 5,0509 | 5,2851 | 72,33% | 75,68% |
| | | 512 | 6,7252 | 4,9155 | 5,1420 | 73,09% | 76,46% |
| | | 128 | 4,9764 | 3,0741 | 3,3001 | 61,77% | 66,32% |

Tabela 13: Comparação, para três dados, entre a entropia após calcular a diferença lateral e a entropia após calcular a diferença seguindo a curva de Hilbert.

Apesar da utilização da curva de Hilbert parecer interessante nossos testes demonstraram que não houve ganhos qualitativos com esta estratégia, fortalecendo nossa idéia original de trabalhar com a diferença lateral que ainda apresenta a vantagem da simplicidade de implementação.

## 3.7. Predição por casamento parcial - PPM

Sabemos que os dados sísmicos possuem coerência espacial e que amostras próximas tendem a ter valores parecidos. Os testes com a diferença lateral provam a existência desta relação. A diferença obtida seguindo a curva de Hilbert também apresentou resultados compatíveis com esta coerência, embora sem ganho sobre a diferença lateral. Para completar o estudo resolvemos avaliar também a freqüência de seqüência de amostras.

Muitos métodos de codificação aproveitam-se das diferentes freqüências de cada símbolo. No método conhecido como PPM, do inglês *Predction by Partial Matching,* em vez de examinarmos os símbolos isoladamente, consideramos uma seqüência deles e o código atribuído a cada um dependerá dos que o precedem.

Quando introduzimos a idéia de olhar para uma seqüência de símbolos para a compressão podemos fazê-lo de mais de uma maneira. A primeira é modificar o dicionário de forma que cada possível seqüência seja considerada

um símbolo. As freqüências são recalculadas e um código é gerado para cada seqüência. O problema desta técnica é que o número de símbolos possíveis cresce exponencialmente com o tamanho da seqüência. Outra maneira é calcular a freqüência condicional de cada símbolo. Isto é, para cada símbolo, calcula-se a probabilidade de ele aparecer dependendo da seqüência anterior a ele.

O PPM é um método adaptativo, não trabalha com uma seqüência de tamanho fixo mas de tamanho variável. Para cada símbolo encontrado o PPM tenta gerar uma codificação examinando os *j* caracteres anteriores, se tal seqüência nunca tiver ocorrido então o contexto é reduzido e os *j-1* caracteres anteriores são observados. Dependendo de quantas vezes o contexto já ocorreu e de quais símbolos vieram seguidos em cada caso, pode-se prever a probabilidade de ocorrência do símbolo que está se querendo codificar e, em função desta probabilidade, gerar um código. Desta possibilidade de casar um contexto completo ou apenas parte dele vem o nome PPM.

Além de ser adaptativo no contexto o PPM também se adapta ao texto como um todo. As freqüências não são pré-calculadas e sim calculadas à medida que o compressor passa pelo texto. Isso permite que a compressão seja feita em uma única passada.

### 3.7.1. Algoritmo

Para entender como funciona o PPM vamos supor que estamos comprimindo um texto e queremos comprimir um símbolo *s* precedido por *j* símbolos aos quais chamamos de contexto. Neste ponto três diferentes situações podem ocorrer: o contexto nunca ocorreu antes, o contexto já ocorreu seguido do símbolo em questão ou o contexto já ocorreu, porém nunca seguido do símbolo em questão. No primeiro caso nada é escrito, e marcamos uma ocorrência deste contexto seguida de *s*. Depois diminuímos o contexto para os *j-1* caracteres que precedem s e repetimos a tentativa de codificação. O segundo caso é o almejado pelo PPM, comparamos o número de vezes que *s* ocorreu neste contexto com o total de vezes que o contexto ocorreu e geramos um código para *s* baseado nesta razão. Quanto mais vezes *s* tiver ocorrido menor será seu código. O terceiro caso é um caso especial do segundo. A ocorrência de *s* no contexto é anotada e teríamos que criar um código para *s* de acordo com sua probabilidade de ocorrer neste contexto, porém sua probabilidade até então

era zero. A solução é indicar ao decodificador que o símbolo não pode ser codificado neste contexto. Um caractere especial é escrito e o contexto é reduzido. A tentativa é repetida, então, no contexto seguinte.

Trata-se de um código recursivo. O tamanho do contexto vai diminuindo até que se chegue a um contexto que já tenha ocorrido seguido de *s*. No começo da compressão é provável que apareçam símbolos nunca antes ocorridos. Daí ele não poderá estar na contagem de nenhum contexto, nem mesmo o contexto zero. Para evitar este problema iniciamos a contagem do contexto de tamanho zero com uma ocorrência para cada símbolo.

Para ilustrar o procedimento vamos imaginar a codificação de algumas palavras ao longo de um texto em português com o PPM onde o contexto máximo tem tamanho dois. Em um determinado momento nos deparamos com a codificação da letra “e” pertencente a palavra “que”. O contexto é então a seqüência “qu”; como esta palavra é comum na língua portuguesa é provável que este contexto já tenha ocorrido e, em grande, parte seguido da letra “e”. Se isto for verdade calculamos a probabilidade da letra “e” ocorrer após “qu” e a codificamos.

Agora suponha que no texto já tenha ocorrido a palavra “que” algumas vezes e surge a palavra “enquanto”. Para codificar a letra “a” examinamos o contexto “qu” que já ocorreu, porém nunca seguido de “a”. Nesse caso não podemos codificar “a” devido a probabilidade zero. O codificador escreve um caractere especial: *ESC*, indicador desta situação e desce para o contexto “u”, se, por exemplo, a palavra “água” já tiver ocorrido então podemos codificar “a” com contexto “u”.

Também podemos imaginar o caso da primeira vez que ocorre a seqüência “qu”, ou seja, ela nunca ocorreu antes. Neste caso a necessidade de descer para o contexto seguinte é implícita e o codificador não precisa escrever o caractere especial. Em todos os casos a atualização do contador só ocorre depois da codificação ou da diminuição do contexto para que codificador e decodificador consigam manter as mesmas tabelas de freqüências ao longo do texto.

### 3.7.2. Experimentos computacionais

O PPM foi testado de quatro maneiras diferentes. Na primeira caminhamos na direção vertical, ou seja ao longo do eixo z, de um dado quantizado a fim de identificar repetições no padrão oscilatório de cada traço. Na segunda caminhamos na horizontal do mesmo dado. Os dois últimos testes comparavam o caminhamento na vertical e na horizontal em um dado após a diferença lateral ter sido calculada.

| Dado | Tipo de Agrupa-mento | K | Entropia PPM vertical | Entropia PPM horizontal | Ent. PPM vertical após diferença | Ent. PPM horizontal após diferença | Entropia após diferença lateral |
|---|---|---|---|---|---|---|---|
| Dado F | Medianas | 128 | 6,581 | 4,796 | 4,865 | 4,835 | 3,941 |
| | | 256 | 7,572 | **5,771** | **5,853** | **5,819** | 5,928 |
| | | 384 | 8,182 | **6,371** | **6,460** | **6,422** | 6,499 |
| | | 512 | 8,605 | **6,785** | **6,875** | **6,836** | 6,921 |
| | Uniforme | 128 | 4,830 | 3,074 | 3,016 | 3,023 | 1,293 |
| | | 256 | 5,837 | 4,055 | **3,991** | **3,993** | 4,054 |
| | | 384 | 6,427 | 4,638 | **4,573** | **4,573** | 4,637 |
| | | 512 | 6,847 | 5,053 | **4,987** | **4,986** | 5,051 |
| Dado C | Medianas | 128 | 5,369 | 3,225 | 2,705 | 2,858 | 2,214 |
| | | 256 | 6,367 | **4,187** | **3,619** | **3,776** | 4,195 |
| | | 384 | 6,981 | 4,785 | **4,207** | **4,364** | 4,781 |
| | | 512 | 7,409 | 5,202 | **4,606** | **4,764** | 5,189 |
| | Uniforme | 128 | 4,238 | 2,191 | 1,748 | 1,890 | 1,073 |
| | | 256 | 5,248 | 3,100 | **2,499** | **2,649** | 3,067 |
| | | 384 | 5,847 | 3,673 | **3,022** | **3,173** | 3,634 |
| | | 512 | 6,278 | 4,090 | **3,414** | **3,566** | 4,042 |
| Dado R | Medianas | 128 | 6,230 | 5,149 | 4,968 | 5,067 | 4,181 |
| | | 256 | 7,309 | 6,211 | **6,022** | **6,121** | 6,198 |
| | | 384 | 7,946 | 6,840 | **6,625** | **6,723** | 6,810 |
| | | 512 | 8,419 | 7,314 | **7,055** | **7,154** | 7,212 |
| | Uniforme | 128 | 3,705 | 2,689 | 2,452 | 2,559 | 1,635 |
| | | 256 | 4,713 | **3,656** | **3,373** | **3,486** | 3,660 |
| | | 384 | 5,310 | 4,245 | **3,945** | **4,060** | 4,239 |
| | | 512 | 5,738 | 4,671 | **4,359** | **4,474** | 4,651 |

Tabela 14: Entropia do PPM com contexto de até 1 símbolo.

| Dado | Tipo de Agrupa-mento | K | Entropia PPM vertical | Entropia PPM horizontal | Ent. PPM vertical após diferença | Ent. PPM horizontal após diferença | Entropia após diferença |
|---|---|---|---|---|---|---|---|
| Dado F | Medianas | 128 | 6,201 | 4,757 | 4,613 | 4,781 | 3,941 |
| | | 256 | 7,431 | **5,797** | **5,642** | **5,805** | 5,928 |
| | | 384 | 8,265 | **6,471** | **6,298** | **6,455** | 6,499 |
| | Uniforme | 128 | 4,367 | 3,025 | 2,737 | 2,980 | 1,293 |
| | | 256 | 5,441 | **4,004** | **3,680** | **3,946** | 4,054 |
| | | 384 | 6,109 | **4,599** | **4,260** | **4,530** | 4,637 |
| Dado C | Medianas | 128 | 4,401 | 2,857 | 2,399 | 2,828 | 2,214 |
| | | 256 | 5,678 | **3,817** | **3,193** | **3,757** | 4,195 |
| | | 384 | 6,618 | **4,484** | **3,754** | **4,358** | 4,781 |
| | Uniforme | 128 | 3,246 | 1,937 | 1,634 | 1,851 | 1,073 |
| | | 256 | 4,302 | **2,720** | **2,159** | **2,625** | 3,067 |
| | | 384 | 5,020 | **3,281** | **2,559** | **3,155** | 3,634 |
| Dado R | Medianas | 128 | 5,982 | 5,148 | 4,586 | 5,157 | 4,181 |
| | | 256 | 7,572 | 6,417 | **5,827** | 6,338 | 6,198 |
| | | 384 | 8,423 | 7,195 | **6,599** | 7,029 | 6,810 |
| | Uniforme | 128 | 3,093 | 2,566 | 2,036 | 2,539 | 1,635 |
| | | 256 | 4,152 | **3,532** | **2,774** | **3,480** | 3,660 |
| | | 384 | 4,837 | **4,151** | **3,306** | **4,072** | 4,239 |

Tabela 15: Entropia do PPM com contexto de até 2 símbolos.

De maneira geral melhores resultados são encontrados aplicando o método após calculada a diferença lateral. Os casos onde o PPM apresenta menor entropia são mostrados em negrito na Tabela 14, para PPM de contexto um e na Tabela 15 para contexto dois. Ainda assim há casos em que o PPM não consegue montar uma boa tabela de predição e a entropia final fica pior que o aquela da diferença lateral. Nos testes realizados os agrupamentos com menos representantes geram piores resultados de PPM. Com 128 grupos nenhum dos testes conseguiu superar a entropia atingida com apenas a diferença lateral. Comparando a Tabela 14 com a Tabela 15 acreditamos que, para dados sísmicos o PPM de contexto máximo igual a dois consegue melhores resultados do que o contexto um. Isso prova a existência de uma coerência comportamental no dado sísmico além da encontrada com a diferença lateral. No caso do PPM aplicado na vertical esta coerência é a característica oscilatória do dado.

Este método, porém, é caro em tempo e memória. Seu tempo é influenciado tanto pelo número de símbolos diferentes existentes no conjunto quanto pelo tamanho do contexto máximo. Esta relação restringe a utilização do método que, no nosso estudo, se mostra melhor justamente quando se

aumentam estas variáveis. Para atingir uma diminuição justificável na entropia é possível que o custo de tempo torne impraticável para execução a cada leitura. O método seria mais indicado apenas para o armazenamento. Ainda para a transferência é necessário avaliar, em cada caso, se o tempo acrescido com a compressão supera o tempo ganho com na transmissão.

## 3.8. Observações finais

Neste capítulo vimos diferentes métodos e conceitos sobre a compressão, agora nos resta analisá-los e montar uma estratégia final para a compressão dos dados sísmicos.

A compressão, conforme vista, consiste em duas etapas, a primeira, a qual chamamos de quantização, trata-se da substituição dos valores originais por índices de uma nova tabela de valores. Nesta etapa está associada a perda de precisão e conseqüentemente o grau de erro do método. A segunda etapa é a codificação onde se transforma cada símbolo em um código. É nesta etapa que se comprime, de fato, o dado.

As duas etapas não são totalmente independentes. Embora elas possam ser executadas desta forma, a qualidade da compressão atingida na codificação dependerá da qualidade da quantização. Vimos nas sessões anteriores que a entropia do dado varia em proporção inversa com o erro. Também vimos que a mesma relação erro x entropia ocorre nos dois métodos de quantização apresentados. A diferença entre eles fica no número de grupos que cada método precisa para atingir um determinado ponto desta relação.

Notamos ainda que o crescimento da entropia em função do número de grupos é mantido após aplicarmos os métodos de codificação. Isso quer dizer que nenhum dos métodos de codificação utilizados consegue compensar o aumento da entropia conseqüente do uso de mais grupos. Assim dados agrupados de forma a ter erros menores continuam com a entropia mais alta que dados agrupados com maior fator de erro. Essa propriedade permite, escolher a melhor codificação sem precisar considerar o método utilizado na quantização, sabendo apenas que seu resultado estará limitado pelo erro introduzido na fase anterior.

Em nossos testes não foi executada nenhuma codificação propriamente dita, porém estudamos seu potencial com a entropia. Calculamos a freqüência

de símbolos, seqüências de símbolos e diferença entre símbolos. Desta maneira saberíamos como tratar o arquivo para uma codificação eficiente.

Tentamos aproveitar a coerência espacial do dado para a codificação. Consideramos a diferença lateral, a diferença seguindo a curva de Hilbert e o PPM. Todos os métodos mostraram vantagem se comparados com a não aplicação de nenhum deles. Quando comparados entre si observa-se que a diferença lateral mostrou os resultados mais significativos. Destes três métodos a curva de Hilbert foi o único que não apresentou ganho nenhum sobre os outros. O PPM foi estudado isoladamente e em conjunto com a diferença lateral. Este método, aplicado isoladamente com contexto máximo de tamanho igual a dois conseguiu, em alguns casos atingir entropia menor que a aplicação da diferença lateral, porém se mostrou ainda mais eficiente se aplicado em conjunto com a mesma.

Para completar o estudo e confirmar nossas estimativas propomos e implementamos uma estratégia que junta os métodos estudados. Para a codificação utilizamos o código aritmético seguindo a implementação apresentada por Witten et Al [12,13] cuja taxa de compressão se aproxima bastante da entropia. Para a estratégia precisamos escolher um método de quantização. Como não temos necessidade de limitar o número de grupos nem vamos utilizar os dados em cálculos matemáticos optamos pela quantização por intervalos uniformes cujo algoritmo é mais simples e eficiente. Para a codificação fizemos a seqüência diferença lateral com PPM vertical de contexto máximo igual a dois, referenciado como PPM-2. O PPM também utilizou o código aritmético para a geração dos códigos de cada símbolo. Os resultados podem ser vistos a seguir na Tabela 16 e na Tabela 17 onde comprovamos que a compressão com código aritmético é semelhante a estimativa da entropia. A compressão final ficou dentro do esperado.

| Dado: C; Tamanho Original: 14930496 bytes<br>Agrupamento: 256 Intervalos Uniformes | | | | | |
|---|---|---|---|---|---|
| Estratégia | Tamanho Real após Compressão | Tamanho Estimado pela Entropia | Bits por Amostra após Compressão | Entropia | Razão Comprimido / Original |
| PPM-2 após Diferença Lateral | 1007099 | 1007383 | 2.158 | 2.159 | 6.75% |
| Diferença Lateral | 1453204 | 1430910 | 3.115 | 3.067 | 9.73% |
| PPM-2 | 2072922 | 2007442 | 4.443 | 4.302 | 13.88% |
| Somente agrupamento | 3105587 | 3122780 | 6.656 | 6.693 | 20.80% |

Tabela 16: Resultados da compressão com código aritmético em comparação com as estimativas da Entropia no Dado C.

| Dado: F; Tamanho Original: 66846720 bytes<br>Agrupamento: 256 Intervalos Uniformes | | | | | |
|---|---|---|---|---|---|
| Estratégia | Tamanho Real após Compressão | Tamanho Estimado pela Entropia | Bits por Amostra após Compressão | Entropia | Razão Comprimido / Original |
| PPM-2 após Diferença Lateral | 7714956 | 7687094 | 3.693 | 3.680 | 11.54% |
| Diferença Lateral | 8586598 | 8469610 | 4.110 | 4.054 | 12.85% |
| PPM-2 | 11573397 | 11365824 | 5.540 | 5.441 | 17.31% |
| Somente agrupamento | 12494321 | 12493903 | 5.981 | 5.980 | 18.69% |

Tabela 17: Resultados da compressão com código aritmético em comparação com as estimativas da Entropia no Dado F.