

Métodos de compressão mais usados em codecs para sinais ECG

André Moreira nº 2017275462, José Lamas nº 2017259895, Marcelo Gomes nº 2017278985
DEI, Departamento de Engenharia Informática
FCTUC, Faculdade de Ciências e Tecnologia da Universidade de Coimbra

Abstrat

Neste trabalho fala-se de diferentes métodos de compressão sem perdas (lossless) e compressão com perdas (lossy). Sendo eles a codificação de Huffman e o método da Transformada Discreta do Cosseno (DCT) como métodos com perda. E o Run Length Encoding e o método LZW como métodos sem perda. Irá, também, ser comparado esses métodos quanto à sua taxa de compressão (CR) e quanto ao Erro Quadrático Médio normalizado (PRDN).

1. Introdução

Um eletrocardiograma é o exame cardíaco mais comum, que tem como objetivo analisar o coração e detetar possíveis doenças cardíacas. Visto que os dados recebidos pelo exame podem ser extensos, há uma necessidade de serem comprimidos.

Essa compressão é feita através de codecs. Estes usam diversos métodos/algoritmos para reduzir o número de bits usados para representar um sinal e, ao mesmo tempo, preservar a informação do sinal. Para comprimir uma sequência de dados, o algoritmo deve examinar os dados, achar redundâncias, e tentar tirar essas redundâncias.

Num codec, quando se estabelece um tipo de compressão para usar, precisa-se de saber se, no ato de decodificação, pode existir perdas relativamente ao sinal (compressão do tipo lossy), ou não (compressão do tipo lossless).

Pode-se avaliar o desempenho de um algoritmo, usando a CR (taxa de compressão) e o PRDN (Percentage Root Mean Square Difference).

Quanto a métodos de compressão usados em codecs do tipo lossy, existe, por exemplo, a codificação de Huffman e o DCT (Discreet Cosine Transform). E quanto a métodos de compressão do tipo usados em codecs do tipo lossless, temos RLE (Run Length Encoding) e o LZW (Lempel Ziff Welch).

Neste trabalho irá ser comparado alguns métodos de compressão em codecs de ambos os tipos.

2. Métodos de compressão lossy

2.1. DCT (Discreet Cosine Transform)

O DCT é um método usado na fase de redução da redundância. O DCT (Transformada discreta do cosseno) é muito usado em codecs para comprimir imagens perdendo, no entanto, alguma qualidade.

Existe 8 variantes deste método de compressão, sendo o mais usado o tipo DCT-II, encontrando-se, por exemplo, no codec JPEG, um dos mais usados atualmente. Este divide a imagem em partes tendo em conta a sua importância para a qualidade da imagem e cada parte é transformada através de uma DCT bidimensional.

Como no método DCT existe alguma perda de informação, costuma ser usado com outros métodos (como a codificação de Huffman, codificação aritmética e o Run Length Encoding) para que esta seja menos significativa.

$$F(u, v) = \frac{2c(u)c(v)}{N} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f(m, n) \cos\left(\frac{2m+1}{2N}u\pi\right) \cos\left(\frac{2n+1}{2N}v\pi\right) \quad (11.28)$$
$$u = 0, 1, \dots, N-1 \quad v = 0, 1, \dots, N-1$$

where

$$c(k) = \frac{1}{\sqrt{2}} \text{ for } k = 0 \\ = 1 \text{ otherwise}$$

Figura 1 - Discreet Cosine Transform

2.2. Codificação de Huffman

Este método de compressão, também usado na fase de redução de redundância e entropy encoding, é um dos mais utilizados na atualidade como um componente de codecs mais complexos. Serve para criar códigos de prefixo com base na probabilidade de ocorrências de símbolos na fonte informação a ser comprimida para determinar códigos de tamanho variável.

O algoritmo baseia-se na construção duma árvore binária baseada nas probabilidades de ocorrência de cada símbolo para atribuir as ocorrências mais frequentes. Na árvore, as folhas representam os símbolos presentes nos sinais e a sua probabilidade de

O algoritmo começa por juntar dois símbolos com menor probabilidade, juntando-os em um nó e atribuindo a soma das suas probabilidades. Este nó passa a ser tratado como se fosse um novo símbolo do alfabeto. Após isto vai verificar de novo qual os dois símbolos com menor probabilidade e continua a criar novos nós até haver apenas 1 (a raiz) em que a soma das probabilidades dos seus descendentes é 1.

A 0.4
B 0.2
C 0.2
D 0.1
E 0.1

1
1
1
0
0

0.6
0.4
0.2
0

1.0

A 1
B 01
C 001
D 0001
E 0000

É um método muito simples de compressão do tipo lossless dado que este apenas substitui o caracter que se repete consecutivamente pelo número de vezes que se repete vezes, tendo, normalmente, um símbolo a separar estes dois, sabendo que o mesmo não pertence à cadeia de maneira a distinguir-se dos símbolos que pertencem à cadeia ou se foi introduzida pelo RLE. Por exemplo AAAAAADDEEFE após passar por um

Figura 4 – Exemplo de Codificação LZW

4. Comparação dos métodos

A eficiência dos métodos pode ser avaliada usando a fórmula da taxa de compressão e também, no caso de ser um algoritmo com perda de informação, o Erro Quadrático Médio normalizado (PRD-Percentage Root Mean Square Difference).

$$CR = \frac{nBitsOriginal}{nBitsComprimido}$$

$$PRD = \frac{\sum_{k=1}^N (x(k) - x_{rec}(k))^2}{\sum_{k=1}^N x(k)^2} \times 100$$

Figura 5 – Fórmulas do ratio de compressão e do PRD, respetivamente

A seguinte tabela e informações foi retirada de *. Tem-se como fonte de informação 4 sinais de ECG digitalizados através de amostragem a 720Hz com uma resolução de 12bits.

Digitalized ECG signals @ 720 Hz with 12-bit resolution			
Name	Acquisition	Size	Duration
aami3a	s1.txt	506K	30s.
	l1.txt	0.98M	60s.
aami3b	s2.txt	506K	30s.
	l2.txt	0.98M	60s.
aami3c	s3.txt	506K	30s.
	l3.txt	0.98M	60s.
aami3d	s4.txt	506K	30s.
	l4.txt	808K	60s.

Figura 6 – Tabela com sinais ECG digitalizados usados como teste

Após testar estes métodos de compressão obteve-se os seguintes resultados:

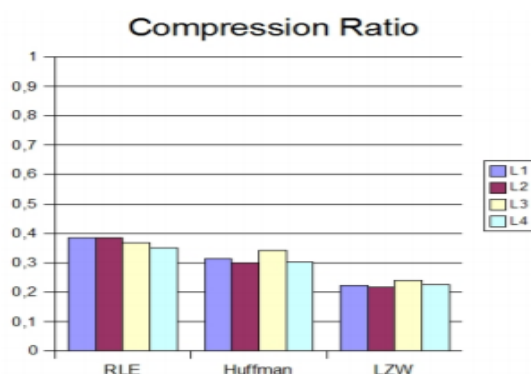


Figura 7 – Gráficos do ratio de compressão dos sinais usando os diferentes métodos

Neste caso, tem-se o RLE com maior taxa de compressão (entre 35 a 40%), seguido das codificações de Huffman (entre 30 a 35%) e finalmente a codificação LZW (entre 20 a 25%). Estes, por serem métodos *lossless*, não tem degradação da informação, ou seja, não há erro quadrático médio (PRD).

5. Conclusão

Neste trabalho foram apresentados alguns dos métodos de compressão usados em codecs para sinais ECG na atualidade. Estes podem ter perda, ou não, de informação. Para aumentar a eficiência de um codec pode-se usar vários métodos/algoritmos nas diversas fases de um codec.

6. Referências

M. Brito, J. Henriques, P. Carvalho, B. Ribeiro, M. Antunes, An Ecg Compression Approach Based On A Segmente Dictionary And Bezier Approximations

Compression Techniques [Internet], Available from: https://www.sqa.org.uk/e-learning/BitVect01CD/page_82.htm

A Comparison of Single Lead Ecg Data Compression Techniques, Available from: [http://www.hcs.harvard.edu/~weber/HomePage/Papers/ECG Compression/](http://www.hcs.harvard.edu/~weber/HomePage/Papers/ECG%20Compression/)

Marcelo Adrián Campitelli, Compressão de Sinais Ecg Utilizando Dwt Com Quantização Não-linear E Por Sub-bandas

Monica Negoita, Liviu Goras, On A Compression Algorithm For Ecg Signals

* Laia Bayarri Portolés, Lossless Compression Of Ecg Signals – Performance Analysis in a Wireless Network –

Scottish Qualifications Authority, F1YX 34: Digital Imaging: Bitmap and Vector

Mário de Andrade Pereira de Brito, Compressão de ECGs. Uma Abordagem Baseada na Regularidade do Sinal.

Dave Marshall, The Discrete Cosine Transform (DCT) Available from: <https://users.cs.cf.ac.uk/Dave.Marshall/Multimedia/node231.html>

Danoja Dias, JPEG Compression Algorithm, Available from: <https://medium.com/@danojadas/jpeg-compression-algorithm-969af03773da>