CODIFICAÇÃO DE ÁUDIO E VÍDEO

Codificador de Áudio

November 12, 2016

Rui Espinha Ribeiro, 68794
André Lopes, 67833

Chapter 1

Introdução

Para efeitos de armazenamento, é bastante comum usar compressão de aúdio para poupar espaço em dispositivos. Nestas técnicas, é fornecido um ficheiro com amostras aúdio quantizadas, neste caso no formato .wav, e procura-se obter um ficheiro mais pequeno que, no caso de compressão *lossless*, não apresenta perdas de informação aquando a sua descodificação. Nesta tarefa procura-se implementar métodos de codificação de aúdio *lossless* e *lossy*, observar a diferença entre estes, a taxa de compressão possível com diferentes técnicas.
Para este trabalho são fornecidos vários ficheiros no formato .wav, onde se pretende ler cada um deles bit a bit, usar um modelo a preditivo linear, onde se procura prever o valor da amostra seguinte. codificamdp a diferença (redundância) entre a estimative o valor real, retirado do ficheiro wav, usando códigos de Golomb. Neste contexto, os modelos preditivos utilizados são apenas aqueles que nos permitam obter valores residuais que se aproximem de uma distribuição geométrica, pois é para este tipo de distribuição que os codificadores de Golomb permitem obter um código prefixo ótimo.
Deste processo resulta um ficheiro, potencialmente, mais pequeno que o original. No caso de compressão lossless, o processo inverso deverá produzir um ficheiro wav idêntico ao original.

Implementação

Leitura *bitwise*

Para a codificação é essencial a manipulação de bits individuais, algo que não é possível directamente com as classes nativas de C++. Para isso, desenvolveu-se um módulo denominado *BitStream* que permite a escrita/leitura de 1 ou N bits num ficheiro.

Encoding e Decoding

Para o encoding são usados códigos de Golomb. Dado um parâmetro M , divide-se um valor de entrada N, obtendo como quociente q e o resto r, q é enviado em código unário, r é enviado em binário truncado. Assim, se M é potência de 2:

$$b = \log_2 M \tag{1.1}$$

Sendo b o número de bits necessários para representar r. Os *bits* correspondentes aos códigos unário e binário são então escritos com a classe BitStream.

Uma das dificuldades encontradas no encoding foi que, frequentemente, o resultado do quociente entre N e M era um número negativo, o que iria resultar num acesso a um índice negativo. Para evitar esta situação transformou-se o valor de entrada N segundo a seguinte regra: Se N >= 0, envia-se 2N para o codificador, caso contrário,envia-se 2|x|-1. Assim, no processo de descodificação conseguimos aferir se o número original é positivo ou negativo verificando se o número que os bits de codificação nos permitem ler é par ou ímpar.

Modelos preditivos usados

Usaram-se modelos preditivos polinomiais simples de profundidade crescente. Estes predictors consideram-se temporais pois o resultado da previsão está dependente de valores anteriores.Inicialmente, idealizou-se o *predictor* mais simples concebível: a amostra seguinte é igual à amostra anterior.
Foi também feito um outro *predictor* com profundidade 2. Dado *samples* como um array contendo as amostras num ficheiro .wav, cuja estimativa é:

```
valor_previsto = 2*valor[i-1] - valor[i-2]
```

Visualização dos Histogramas

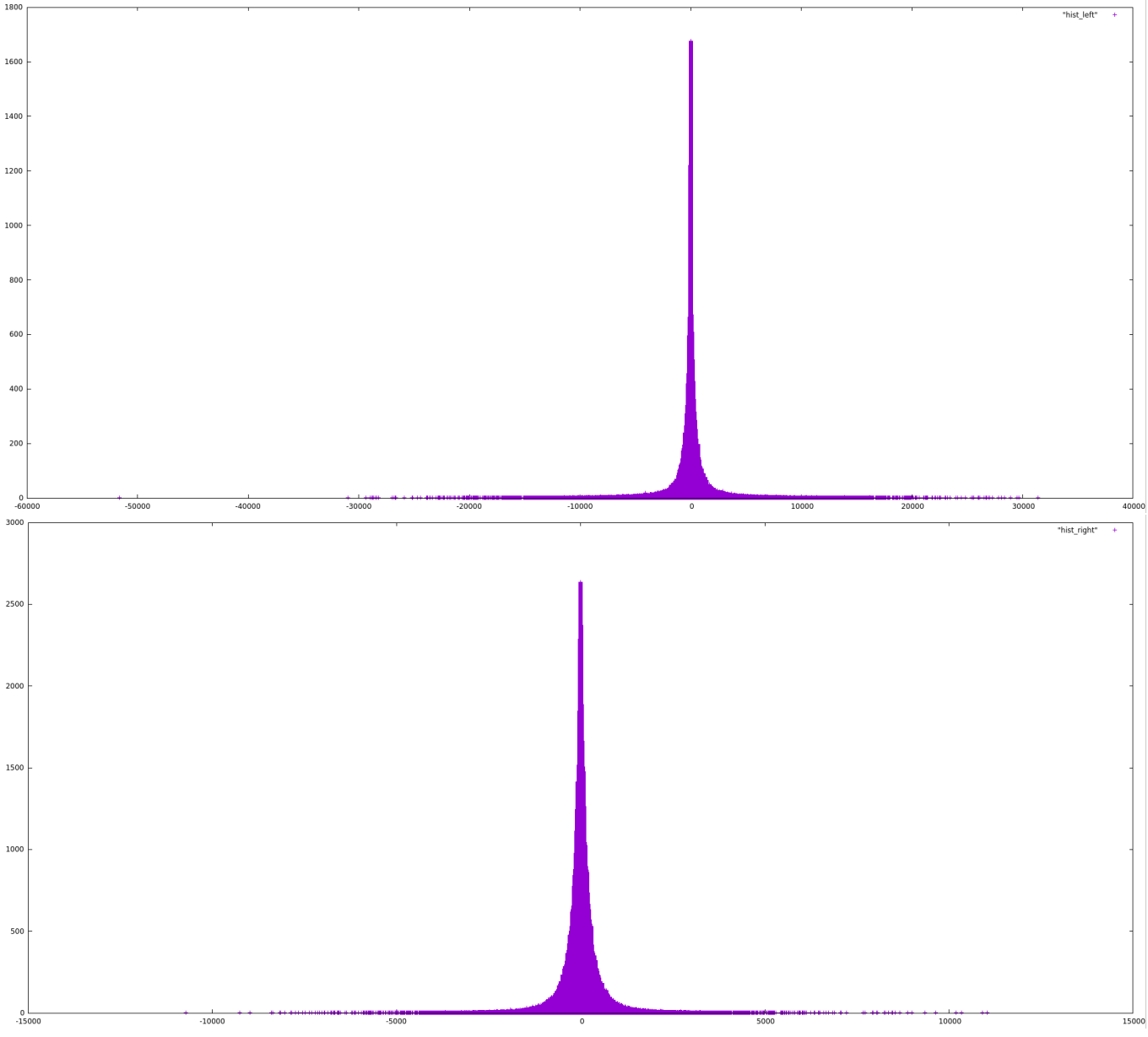
Como base de comparação entre as diferentes distribuições, foram traçados histogramas a partir dos valores residuais enviados para o codificador e calculada a entropia correspondente a cada distribuição. Utiliza-se a ferramenta *gnuplot* para a visualização do histograma.

Logicamente, o histograma ideal será aquele em que se verificar a maior convergência possível dos valores residuais em zero, o que significa que a estimativa do modelo preditivo é mais precisa e são codificados assim valores mais pequenos (o que por sua vez implica menos *bits* de codificação.

Testes

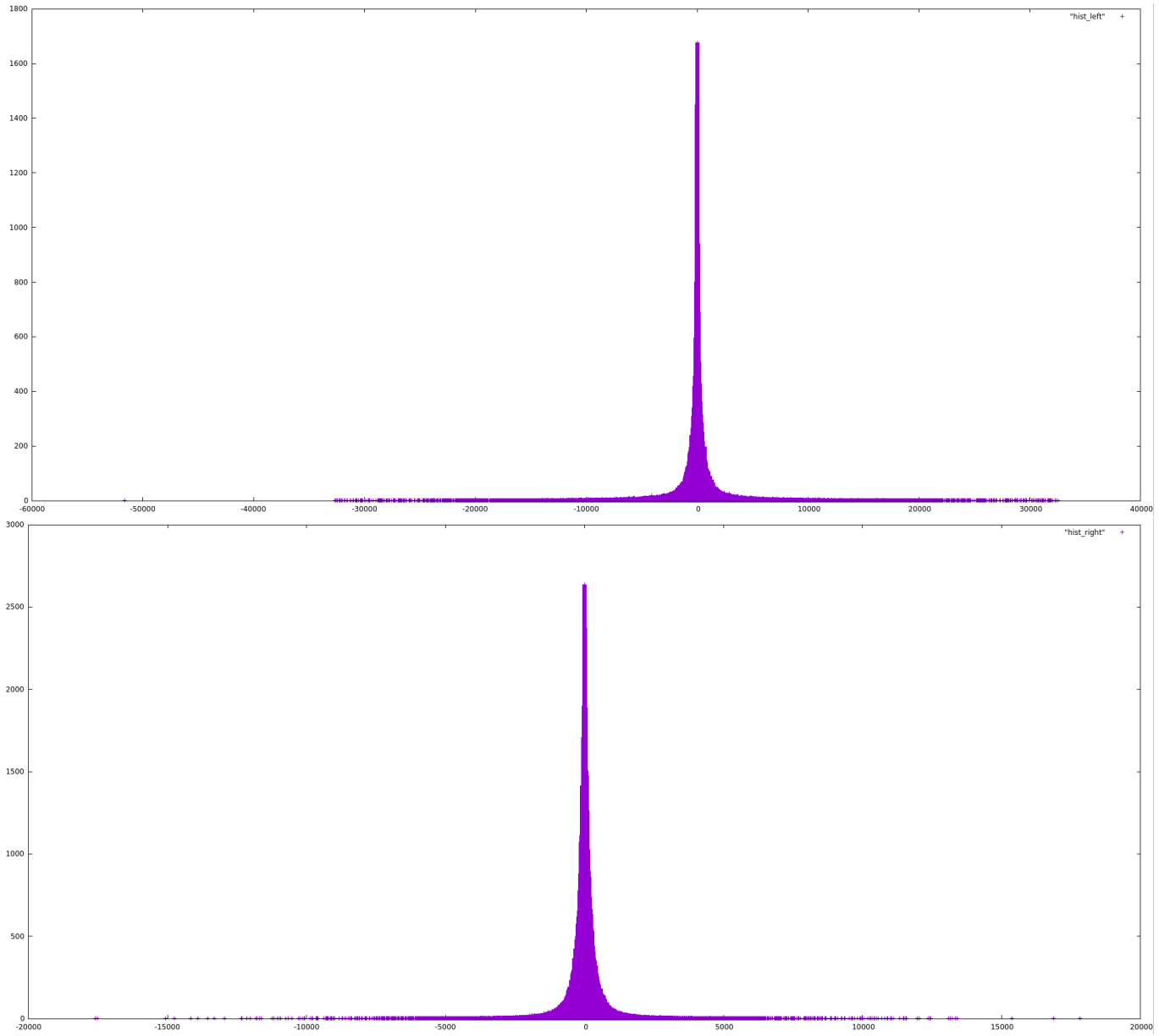
Histogramas dos valores residuais

Modelo de profundidade 1



Entropia Canal 1: 239.29 | Canal 2: 366.116

Modelo de profundidade 2



Entropia Canal 1: 205.24 | Canal 2: 326.478

Foram feitos testes com diferentes valores do parâmetro *M* para os dois diferentes modelos. Foi possível verificar que para valores maiores de M, o tempo de descompressão era maior. Com um factor de divisão maior, o valor do quociente é naturalmente mais pequeno, sendo assim escritos menos bits pelo codificador de Golomb.

Discussão

Conclusões

Os histogramas gerados foram de encontro aos resultados previstos, observando-se uma distribuição geométrica dos valores residuais, existindo uma convergência em zero. O cálculo da entropia também revela o que já era um pressoposto teórico: quanto maior a profundidade, menor a entropia pois mais valores existentes são tidos em conta para gerar a previsão.

Foi possível codificar e descodificar ficheiros áudio com esta implmentação. No entanto, os tempos de descompressão são muito altos e as taxas de compressão não são satisfatórias (observamos expansão ao invés de compressão). O segundo problema seria potencialmente ultrapassado com a implementação de modelos preditivos, que permitam gerar valores residuais ainda mais convergentes em zero, para existirem valores mais pequenos a codificar. Tendo um número de frames na ordem dos milhares ou milhões para um ficheiro .wav, aplicando codificadores de Golomb, o número vai ter sempre o Q+1+B bits, em que N é o valor do quociente e o B é o número de bits necessário para codificar o resto da divisão. Assim, com vários valores residuais na ordem das centenas ou milhares, podemos acabar a utilizar mais bytes para a codificação do que o número de bytes original do valor original.

Possíveis melhorias

Não foi implementado nenhum modelo preditivo que utilize redundância dos canais do ficheiro áudio, apenas redundância temporal. Este modelo permitia-nos ter mais codificadores diferentes com taxas e tempos de compressão diferentes.
Também não foi implementada codificação *lossy*, o que nos permitia potencialmente verificar taxas de compressão superiores a *lossless*, ainda que não conseguindo descodificar o ficheiro e obter o ficheiro exatamente original, evidenciando perda de informação.