

CODIFICAÇÃO DE ÁUDIO E VÍDEO

Codificador de Vídeo

January 10, 2017

Rui Espinha Ribeiro, 68794

André Lopes, 67833

Chapter 1

Introdução

As técnicas de compressão de vídeo são fundamentais na realidade atual, onde utilizamos vários formatos diferentes de vídeo para utilização doméstica ou profissional. Nesta área, há sempre um *trade-off* no que toca a compressão *lossy* - entre a compressão do vídeo e a qualidade da informação apresentada. Os projetos desta UC procuram explorar este equilíbrio com técnicas de complexidade reduzida em relação ao *state of the art* dos codecs existentes atualmente, tanto em áudio como em vídeo.

Para este trabalho, recorreu-se mais uma vez a modelos de predição com códigos de Golomb para codificação de informação, de modo a tentar obter codecs com e sem perdas.

Foram utilizados ficheiros de vídeo *raw*, cujas *frames* que são analisadas tanto em comparação espacial como temporal dos *pixels*.

Implementação

Codificação Intraframe

Nesta abordagem, recorremos a predição espacial para estimar valores para cada *pixel* com base na sua vizinhança. De entre os modelos fornecidos na UC, foi considerado para o trabalho o mais eficiente, o JPEG-LS:

$$\hat{x} = \begin{cases} \min(a, b) & \text{if } c \geq \max(a, b) \\ \max(a, b) & \text{if } c \leq \min(a, b) \\ a + b - c & \text{otherwise} \end{cases}$$

O vídeo *raw* é analisado *frame a frame*, sendo codificados os valores residuais resultantes da diferença entre o valor real do *pixel* e a estimativa do preditor, obtendo assim um compressor *lossless*. Foram obtidas taxas de compressão próximas dos 56% (de 200MB para 88MB no primeiro vídeo e de 214 para 95MB no segundo vídeo fornecido). É possível testar este codec *lossless* com os programas SpatialEncoder e SpatialDecoder. O decodificador gera um ficheiro .rgb, de modo a conseguir aferir a consistência em relação ao ficheir original. Através de testes com vários valores do parâmetro M (Golomb), foi considerado ideal o valor 4.

Codificação Interframe

Neste *codec*, é utilizado um modelo de predição temporal, baseado em comparação entre *frames* consecutivas. Cada *frame* é dividida em blocos de dimensões parametrizáveis e é codificada a diferença entre cada *pixel*. A primeira *frame* é naturalmente codificada com o preditor espacial desenvolvido anteriormente.

Este modelo provou ser menos eficaz para compressão sem perdas, possivelmente devido facto de haver uma diferença significativa entre *frames* consecutivas (vídeo em constante movimento). Para o caso do primeiro ficheiro a taxa de compressão foi de 44% e para o segundo foi de 48%.

Para este caso, o valor M considerado como ideal foi de 15.

Codificação Híbrida

Este codificador procura tirar partido dos dois anteriores para conseguir obter um *codec* mais eficaz. *Frame a frame*, é feita uma análise do residual médio obtido por comparação espacial

e temporal e é selecionado o método com o valor menor, sinalizando-o em cada frame com um bit.

Em harmonia com os resultados previamente obtidos, este *codec* acaba por "escolher" sempre o codificador espacial. Assim, as taxas de compressão são exatamente as mesmas que no primeiro caso.

Considerou-se desenvolver uma extensão deste codificador *lossless*, que verifica o erro médio bloco a bloco dentro da mesma *frame*, podendo ser útil dado que há blocos com menos alterações no tempo do que outros num vídeo com movimento.

Codificação *lossy* e melhorias ao trabalho desenvolvido

Foram experimentadas algumas abordagens de codificação com perdas, consistindo em alterações aos *codecs* desenvolvidos nas fases anteriores. Inicialmente, procurámos definir um *threshold* arbitrário, abaixo do qual todos os residuais seriam codificados com o valor 0, sendo considerados pouco significativos. Ao aplicar este método com códigos convencionais de Golomb sobre cada *pixel* sobre valores muito baixos de cada residual obtido, não só o vídeo decodificado não tinha qualidade suficiente para ser visualizado corretamente como a própria taxa de compressão verificou-se demasiado baixa para podermos considerar esta experiência bem sucedida.

Alterativamente, procurámos codificar valores residuais acima de um dado *threshold* com o próprio valor de *threshold*, procurando minimizar os bits introduzidos no ficheiro comprimido. Conseguimos visualizar vídeos com imagens de qualidade razoável, sendo o vídeo bastante perceptível. No entanto, também não se verificou uma taxa de compressão minimamente significativa. Pensámos numa alterantiva futura a esta estratégia que codificava os residuais "artificiais" de forma mapeada, como acontece em *codecs* como o LZ77.

Poderíamos também ter desenvolvido um codificador que aplicasse *subsampling* às imagens, transformando a imagem RGB em YCbCr 4:2:0, de modo a procurar codificar mais a informação que a luminância. Isto iria introduzir perdas quantizáveis nas imagens e juntamente com um codificador híbrido mais eficaz, poderíamos ter obtido um produto final mais interessante. Também poderíamos ter procurado um modelo de *motion compensation*, procurando o melhor bloco para codificação no método temporal.

Dado não haver nenhum método de compressão *lossy* na versão final do *software*, não colocámos imagens de experiências no relatório, dado que os *codecs* apresentados para testes são

apenas *lossless*, logo o vídeo apresentado no ficheiro decodificado é exatamente igual ao original.