Image and Video Compression

Licenciatura em Engenharia Informática

Complementos sobre Linguagens de Programação

Docentes:

António J. R. Neves Armando J. Pinho Lúcia Sousa

Alunos:

Bárbara Nóbrega Galiza – 105937

Tomás António de Oliveira Victal - 109018

João Miguel Dias Andrade - 107969

Janeiro 2024

Conteúdo

1 Introdução

Esse relatório tem como objetivo apresentar os resultados obtidos com o projeto de compressão de imagens e vídeos, em especial no que toca ao codec de vídeo desenvolvido.

Nesse sentido, o documento foi divido em duas secções: uma para a codificação com perdas (”lossy”), e outra para a codificação sem perdas (”lossless”). Em cada uma delas, explicamos sua implementação e apresentamos os tempos de execução, taxas de compressão e gráficos que ilustram alguns dos resultados obtidos.

2 Lossless

2.1 Implementação

Para desenvolver um encoder híbrido, utilizamos uma combinação entre um codificador intraframe e um codificador interframe.

No codificador intraframe, para fazer a codificação, utilizamos o JPEG-LS para gerar um valor estimado, do qual utilizamos para calcular os residuais. Depois, codificamos esses residuais usando os códigos Golomb.

Para descodificação, fazemos o processo inverso.

No codificador interframe, dividimos o frame em blocos e procuramos o bloco mais parecido no frame anterior para cada um dos blocos do frame atual, dentro da área de procura (”search area”). Então, guardamos o vetor de movimento para esse melhor bloco (”best block”) e os residuais entre os 2 blocos.

No decoding, buscamos o bloco do frame anterior a partir dos vetores e adicionamos os residuais.

2.2 Testes

Para definir os parâmetros que geram os melhores resultados, fizemos diversos testes. Esses testes foram feitos separadamente para cada parâmetro, em que o melhor resultado do anterior era utilizado no posterior.

No caso do lossless, testamos os seguintes parâmetros:

- Constante do Golomb
- Search area
- Periodicidade

2.2.1 Golomb

Original File	Encoded File	Golomb Value	Time Taken (seconds)
1318.36MB	1098MB	8	143
1318.36MB	1018MB	16	140
1318.36MB	1074MB	32	140
1318.36MB	1194MB	64	144
1318.36MB	1340MB	128	147

Tabela 1: Resultados com a variação da constante do Golomb

2.2.2 Search area

Original File	Encoded File	Search Area	Time Taken (seconds)
1318.36MB	1018MB	5	150
1318.36MB	1019MB	10	322
1318.36MB	1020MB	15	616
1318.36MB	1020MB	20	1018

Tabela 2: Resultados com diferentes search areas

2.2.3 Periodicidade

Original File	Encoded File	Periodicity	Time Taken (seconds)
1318.36MB	1018MB	2	140
1318.36MB	1021MB	4	154
1318.36MB	1022MB	6	158
1318.36MB	1023MB	8	162
1318.36MB	1023MB	10	165

Tabela 3: Resultados com periodicidades diferentes

2.3 Valores finais

- Ficheiro original: 1318.36MB

- Ficheiro codificado: 1018MB
- Taxa de compressão: 1.2950
- Tempo de execução: 140 segundos
- Constante do Golomb: 16
- Search area: 5
- Periodicidade: 2

3 Lossy

3.1 Implementação

Para permitir o lossy encoding, modificamos apenas o codificador interframe. Nesse sentido, utilizamos os argumentos do quantization steps para dividir o espetro de valores possíveis em n subgrupos, em que n é o número de steps, e buscamos o valor inicial de cada subgrupo para ser o novo valor que será guardado nos residuais. Desse modo, passam a existir um menor número de possibilidades para os valores presentes nos residuais.

3.2 Testes

Para os testes de parâmetros, em um primeiro momento, procedemos de forma similar ao que foi feito no caso lossless, com a diferença da introdução de um novo parâmetro, a quantização (”quantization”), e da comparação de tamamhos de blocos. Estes foram os resultados:

Original File	Encoded File	Golomb Value	Time Taken (seconds)
263.672MB	219MB	8	29
263.672MB	204MB	16	29
263.672MB	215MB	32	28
263.672MB	239MB	64	30
263.672MB	268MB	128	29

Tabela 4: Resultados com a variação da constante do Golomb

Com isso, concluímos que 16 era o melhor valor para a constante.
Concluímos que o melhor blockSize era 32.

Original File	Encoded File	Block Size	Time Taken (seconds)
263.672MB	250MB	2	49
263.672MB	217MB	4	35
263.672MB	208MB	8	30
263.672MB	204MB	16	29
263.672MB	203MB	32	27

Tabela 5: Resultados para tamanhos diferentes dos blocos

Para a search area, decidimos utilizar um valor que não fosse tão grande, pois isso aumentava o tempo de execução, e por isso escolhemos 3.

Em seguida, testamos a variação da periodicidade em conjunto com a quantização, por saber que estes valores se relacionam. Obtivemos os seguintes resultados (foram obtidos a partir do vídeo ”ducks_take_off”cortado para só conter os 3 segundos iniciais):

Compression Ratio	Periodicity	Quantization	Time Taken (seconds)
1.39522	2	2	16
1.35543	2	4	17
1.33704	2	8	17
1.32751	2	16	17
1.32124	2	32	17
1.30322	2	64	19
1.29872	2	128	20
1.41307	4	2	17
1.36338	4	4	17
1.34074	4	8	17
1.32906	4	16	18
1.32138	4	32	18
1.2996	4	64	20
1.29418	4	128	21
1.41951	6	2	17
1.36646	6	4	18
1.34244	6	8	17
1.33002	6	16	18
1.32192	6	32	19
1.29878	6	64	20
1.29302	6	128	22
1.42348	8	2	16
1.36846	8	4	17
1.34352	8	8	17
1.3307	8	16	18
1.32225	8	32	18
1.29839	8	64	21
1.29242	8	128	21
1.42608	10	2	17
1.36967	10	4	18
1.34422	10	8	18
1.33109	10	16	18
1.32246	10	32	19
1.29799	10	64	21
1.29194	10	128	21

Tabela 6: Resultados com a variação da quantização e periodicidade

Portanto, concluímos que a periodicidade de 10 em conjunto com a quantização trouxe os melhores resultados.

3.3 Valores finais

Assim, produzimos um gráfico para demonstrar a variação da taxa de compressão conforme a taxa de qualidade, dada pelo valor da quantização:

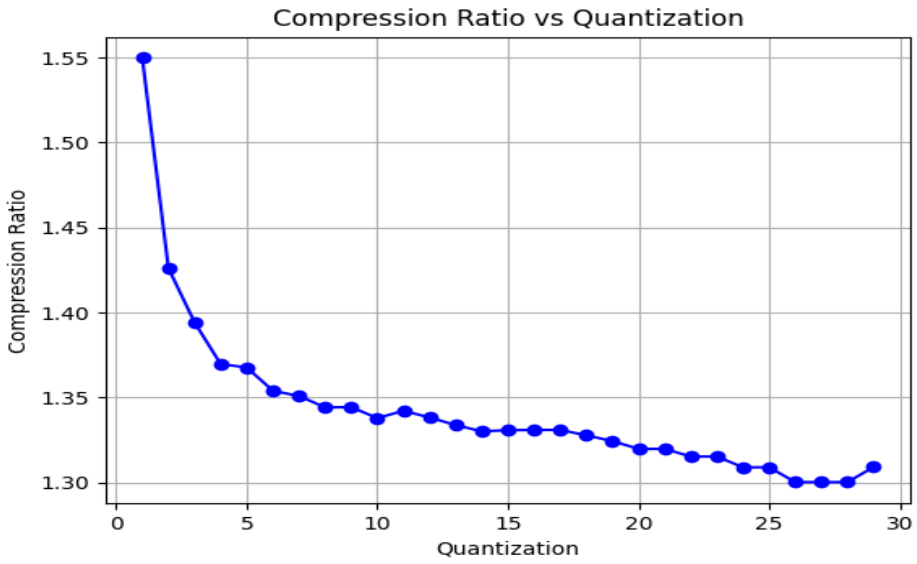

Figura 1: Taxa de qualidade para cada valor de compressão.

4 Conclusão

A realização desse projeto nos permitiu aprofundar os conhecimentos sobre codificação/descodificação, compressão de imagens e vídeos e sobre a linguagem C++.

Em principal, aprendemos em detalhe sobre a codificação por entropia, códigos Golomb, o funcionamento do JPEG-Ls, a diferença entre codecs com perdas e sem perdas e como manipular ficheiros de imagem e vídeo diretamente em C++.

Além disso, concluímos que o tema de compressão de imagens e vídeos traz consigo diversos variáveis e parâmetros que podem ser otimizados, e por isso é preciso uma análise cuidadosa para definir a melhor forma de concretiza-lo.