

Relatório nº3 IC

Edgar Sousa(98757),João Castanheira(97512),
Nuno Fahla(97631)
Turma P1



Conteúdo

1	Introdução	1
2	Ferramentas utilizadas e recolha de resultados e metodologia	2
3	Resumo das funcionalidades implementadas	3
4	Descrição e resultados das funcionalidades implementadas	4
4.1	Exercício 1	4
4.2	Exercício 2	6
4.3	Exercício 3	7
4.4	Conclusão	9
5	Contribuições	10

Capítulo 1

Introdução

Neste trabalho prático para a disciplina de IC, foi-nos pedido para, com base na biblioteca **opencv**, a qual é uma biblioteca para C com o objetivo de ler e manipular ficheiros que contenham amostras de imagens, desenvolver um codec de vídeo, numa primeira fase inteiramente em modo intra-frame, um modo inter-frame e “lossless”, e por fim um modo ‘lossy’ com quantização dos residuais.

Capítulo 2

Ferramentas utilizadas e recolha de resultados e metodologia

Para a execução e análise deste trabalho, foram utilizados a linguagem C++ e as bibliotecas opencv e sndfile. Foram também utilizados os seguintes recursos:

- **time** - comando linux que permite obter o tempo de execução de um programa ou comando-

Neste trabalho prático usamos também a classe BitStream, desenvolvida no guião anterior, de maneira a gerar os ficheiros binários necessários para execução dos CODEC.

Usamos também o software VLC Media player para suportar e analisar todos os formatos de vídeo. E Utilizamos a biblioteca OpenCV apesar das eventuais perdas que irão ocorrer na utilização de esta biblioteca com o formato de vídeo fornecido.

Capítulo 3

Resumo das funcionalidades implementadas

Com a realização deste trabalho prático, as funcionalidades desenvolvidas foram as seguintes:

- Implementação de um codec com Modo puramente intra nomeado Video-EncoderIntra
- Implementação de um codec Lossless com os modos intra e inter por bloco
- Implementação de um codec Lossy com os modos intra e inter por bloco
- Implementação do programa video_cmp

Capítulo 4

Descrição e resultados das funcionalidades implementadas

4.1 Exercício 1

O primeiro exercício deste guião pedia a implementação de um codec de video sem perdas utilizando apenas intra-frame encoding. Os resultados de compressão obtidos podem ser observadas na seguinte tabela.

Tabela 4.1: Tabela com os valores de Compressão Em modo Intra apenas

Amostra	Original	Comprimido	Ratio
akiyo	45621044	26007953	0,5701
bowing	45621044	25593077	0,5601
bridge_close	304140044	212033040	0,6972
bus	22810538	17011913	0,7458

Existe notoriamente alguma perda de qualidade devido á conversão para BRG de YrCrB evidente nas seguintes imagens. Isto pode ser visto nas seguintes imagens.

A metodologia para alcançar o programa é através da iteração pixel a pixel e calculo através da formula: A equação de previsão é descrita na equação seguinte

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

-

Figura 4.1: Efeito da perda de qualidade com o OpenCV

(a) lossless.jpg



(b) Image1.jpg



Onde a corresponde ao píxel diretamente à esquerda do atual b ao diretamente superior e c simultaneamente á esquerda e superior. Como evidente, devido ao uso de esta fórmula, os valores dos pixeis nas arestas da esquerda e superior mantêm-se inalterados. Este fator tornar-se-á relevante no exercício 2.

4.2 Exercício 2

Neste exercício era pedido a adaptação do codec desenvolvido no exercício anterior para a implementação de capacidade de codificação inter-frame. Toda a codificação neste exercício é feita a partir de blocos. A imagem inicial é repartida em blocos inicialmente e caso não seja KeyFrame os blocos são codificados consoante o comprimento da totalidade do bloco codificado, este é o critério de escolha entre os métodos de codificação. No entanto, a codificação bloco a bloco reduz em abundância a eficiência da codificação intra-frame visto que o número de pixels cujos valores não vão ser alterados irá ser muito superior. Outro aspeto a tomar em conta é que o valor search area é um inteiro que representa o raio á volta de um bloco. Outro aspeto a tomar em conta é que os valores de blockSize aceitáveis são apenas valores divisíveis por ambos os valores da resolução do vídeo. Logo o valor de block size máximo será 32. O método de codificação de cada block está dependente do primeiro bit sendo este a 'flag' de cada bloco.

Na tabela seguinte podemos ver o ratio de compressão consoante o tamanho dos blocos.

Tabela 4.2: Tabela com os valores de Compressão com 10 de periodicidade e 1 de search range

Amostra	8	16	32
akiyo	0,7135	0.6147	0,4876
bowing	1.3676	1.1007	0.8309
bus	1.284	1.0258	0.8790

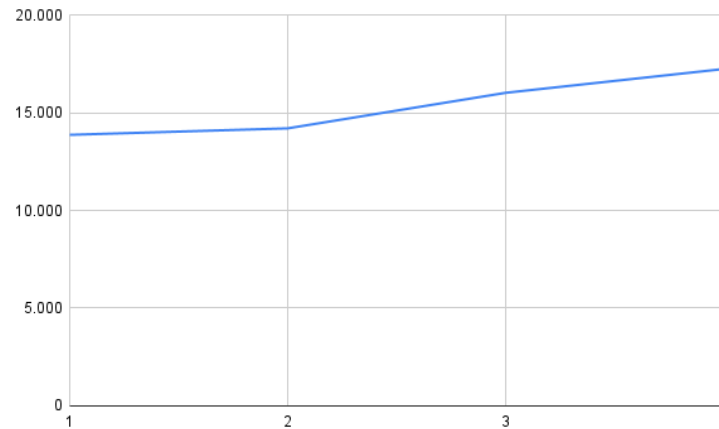
Como podemos observar quanto maior o blocksize melhor o rácio de compressão, e para alguns vídeos a inclusão de motion compensation não é justificável. Poderíamos obter resultados mais satisfatórios com o aumento das dimensões do bloco, no entanto, o programa desenvolvido, como mencionado previamente, aceita apenas dimensões de bloco que são divisores de ambos os valores da resolução. Em retrospectiva, esta limitação é muito impactante, a implementação de "bit Stuffing" seria das primeiras funções, na continuação deste trabalho.

Na seguinte tabela temos o rácio de compressão segundo a search area e a seguir o gráfico de tempo de execução para dimensão de bloco de 32.

Tabela 4.3: Tabela com os valores de Compressão com 10 de periodicidade e dimensão de bloco de 32

Amostra	1	2	3	4
akiyo	0,4876	0.4876	0,4876	0,4876
bowing	0.8309	0.8309	0.8309	0.8309
bus	0.8790	0.8790	0.8790	0.8790

Figura 4.2: process time by search radius



Como podemos observar o aumento da search area não afeta muito positivamente o ratio de compressão mas afeta negativamente o tempo de execução para os videos testados.

4.3 Exercício 3

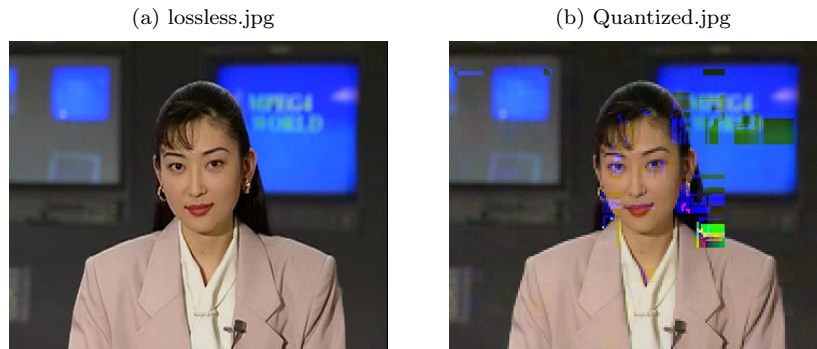
No exercício 3 era requerido a adaptação do codec anterior para implementar um codec lossy através de quantização dos residuais. No entanto os resultados foram insatisfatórios e podem ser evidentes na seguinte tabela.

Tabela 4.4: Tabela com os valores de Compressão segundo quantidade de quantização com 10 de periodicidade e dimensão de bloco de 32

Amostra	1	2	3	4
akiyo	0,4876	0.4855	0.4810	0.4824
bowing	0.8309	0.8284	0.8215	0.8247
bus	0.8790	0.8701	0.8567	0.8555

O resultado visual da quantização podem ser visto nas seguintes imagens.

Figura 4.3: Antes e depois da quantização



Os valores médios de PSNR do programa video_cmp podem ser observados na seguinte lista.

1. akiyo:19.8854
2. bowing:10.3157
3. bus:9.1293

Tanto pela comparação visual como pelos valores apresentados anteriormente, a perda de qualidade introduzida pela quantização é significativa. Enquanto que os ganhos em termos de ratio de compressão não são significativos.

4.4 Conclusão

Toda a execução do trabalho prático, permitiu que desenvolvêssemos capacidade de manipulação de Codecs de vídeo. Apesar da obtenção de resultados não totalmente satisfatórios, aprendemos os métodos utilizados para compressão de ficheiros de vídeo. A ausência de bitStuffing e a utilização de OpenCV seriam dos primeiros aspetos a corrigir no futuro.

Link do repositório com o código desenvolvido

Capítulo 5

Contribuições

1. Edgar Sousa 33%
2. João Silva 33%
3. Nuno Fahla 33%

Acrónimos

IC Informação e Codificação

CODEC Encoder/Decoder