

# Deliverable 4 – Intra & Hybrid Video Codec

Gonalo Cardoso  
85120  
[gnvcardoso@ua.pt](mailto:gnvcardoso@ua.pt)

Miguel Carvalhosa  
84774  
[miguelcarvalhosa@ua.pt](mailto:miguelcarvalhosa@ua.pt)

Tnia Ferreira  
84728  
[tania.s.ferreira@ua.pt](mailto:tania.s.ferreira@ua.pt)

Departamento de Eletrnica, Telecomunicaes e Informtica (DETI)

Universidade de Aveiro, Aveiro

**Resumo**— Este documento descreve a implementao utilizada para desenvolver codecs de vdeo lossless e lossy baseados em codificao intra e intra+inter. Apresenta tambm uma anlise do efeito da variao dos diferentes parmetros do codec na taxa de compresso que permite determinar o conjunto de parmetros timos a utilizar durante a compresso dos vdeos de teste.

**Keywords**—Video compression, lossy, lossless, intra coding, inter coding, hybrid coding

## I. IMPLEMENTAO

Para desenvolver um codec de vdeo *lossless* baseado na codificao de frames intra, foram implementados oito preditores espaciais: os 7 preditores lineares do JPEG e o preditor do JPEG-LS. Estes preditores so essenciais sendo eles que permitem fazer uma compresso de um vdeo, utilizando para isso a redundncia espacial que existe entre pixis prximos de um dado frame. Assim, foram codificados apenas os residuais que resultam da diferena entre o valor original do pixel e o valor resultante da predio ao invs de se codificar diretamente o valor do pixel.

Para garantir que todas as operaes efetuadas na codificao podiam ser invertidas durante a decodificao, foi necessrio especificar no cabealho do ficheiro comprimido o valor dos parmetros utilizados. Estes parmetros incluem o valor de  $m$  utilizado durante a escrita no ficheiro comprimido e o tipo de preditor utilizado para produzir as estimativas e assim, calcular o valor do residual. Desta forma foi possvel obter durante a decodificao a mesma estimativa utilizada durante a codificao de um dado pixel, permitindo assim recuperar o seu valor original.

Dado que se pretende minimizar o tamanho do ficheiro comprimido foi desenvolvido um mtodo que determina de uma forma dinmica o valor timo do parmetro  $m$  do Golomb encoder. O valor de  $m$  utilizado durante a codificao de um conjunto de *estimation\_block\_size* pixis foi estimado com base nos *estimation\_block\_size* residuais anteriores, utilizando-se apenas o valor do parmetro *initial\_m* introduzido pelo utilizador na codificao dos *estimation\_block\_size* pixis iniciais.  importante realar tambm que o Golomb encoder implementado utiliza um cdigo binrio truncado que lhe permite funcionar com qualquer valor de  $m$  estimado, inclusive aqueles que no so potncia de 2, contribuindo ainda mais para minimizao do tamanho do ficheiro comprimido.

Desenvolvida a verso *lossless*, procedeu-se  introduo de perdas nos residuais produzidos no codec anterior atravs da aplicao de um processo de quantizao que descarta os  $n$  bits menos significativos dos mesmos.  importante notar, no entanto, que o mero descarte destes bits provoca uma diferena entre as estimativas produzidas na codificao e decodificao que inevitavelmente faz com que o vdeo decodificado esteja errado. Para garantir a conformidade das estimativas produzidas pelos dois processos, foram utilizados os valores reconstrudos do pixel para produzir a estimativa na codificao ao invs de se utilizar o valor original.

O processo de decodificao implementado neste codec  semelhante ao descrito na sua verso *lossless* com a diferena de se ter de fazer uma sequncia de  $n$  shifts  esquerda aos residuais decodificados antes de se efetuarem as suas somas com as estimativas.

Para desenvolver um segundo codec de vdeo baseado em codificao hbrida foi necessrio adicionar ao codec anterior, que opera exclusivamente em modo intra, um mtodo que permitisse tambm codificar frames em modo inter, i.e, que utilize tanto redundncia espacial como temporal. Para isso, comeou-se por dividir, para os trs planos de cor, a frame em macro blocos de tamanho *block\_size*  $\times$  *block\_size*.

De seguida, para cada macro bloco de cada uma das componentes de cor, procurou-se numa vizinhana de tamanho *search\_area*  $\times$  *search\_area* em torno do bloco da frame anterior, o bloco que minimiza o erro mdio entre o valor dos pixis do bloco que se quer codificar e o valor presente na referncia.

Aps a determinao do bloco de referncia que dever ser utilizado durante a codificao, determinou-se o valor dos vetores de movimento, i.e., a diferena entre a posio do bloco que se quer codificar e a do bloco de referncia para componente de cor. A utilizao destes vetores, conjuntamente com os residuais associados a componente, permitiu codificar cada bloco na ntegra. Assim, foi possvel codificar de uma forma sequencial todos os blocos de uma dada frame.

O processo de decodificao  assim responsvel por ir buscar os blocos de referncia utilizados para codificar as trs componentes de cor  frame anterior nas posies corretas. Estas posies so calculadas fazendo-se a diferena entre as coordenadas do bloco que se quer decodificar e as coordenadas do respetivo vetor de movimento. A soma do valor dos pixis

destes blocos referência aos respectivos residuais decodificados para cada componente de cor permitem recuperar os valores originais. Assim, é possível decodificar de uma forma sequencial todos os blocos de uma dada frame.

Para implementar um método de codificação híbrido, foi necessário adicionar um parâmetro *intraFramePeriodicity*, que indica de quantas em quantas frames deverá ser introduzida uma frame em modo intra.

Por fim, desenvolveu-se a versão *lossy* do codec baseada em codificação híbrida. Tal como aconteceu no caso do codec *lossy* que opera apenas em modo intra, as perdas foram introduzidas nos residuais através do descarte dos  $n$  bits menos significativos dos mesmos. Para garantir novamente que as estimativas feitas durante o processo de codificação eram iguais às da decodificação atualizaram-se (após a codificação de uma frame inteira) os valores da frame anterior para serem iguais aos que vão ser decodificados com erros pelo decodificador, ao invés de se igualarem aos da frame original.

O processo de decodificação implementado é novamente similar ao descrito na versão *lossless* do codec com a diferença de se ter de fazer uma sequência de  $n$  shifts à esquerda aos residuais decodificados antes de se efetuarem as suas somas com os valores dos blocos de referência para cada uma das componentes.

## II. ESTUDO DO EFEITO DOS PARÂMETROS

Nesta secção foi feito um estudo do impacto dos diversos parâmetros do codec na performance da compressão e descompressão. Nos pontos abaixo encontram-se os resultados obtidos na versão híbrida do codec com e sem perdas, variando um parâmetro de cada vez. Em cada teste verificou-se qual o melhor parâmetro e usou-se o resultado nos testes seguintes.

De notar que nos testes foram utilizados vídeos YUV4MPEG2 formatados em 4:4:4. No entanto, o codec tem uma fase de pré-processamento em que os vídeos são convertidos para o formato 4:2:0, pelo que a taxa de compressão é calculada tendo por base um vídeo original em formato 4:4:4 e um vídeo descomprimido em formato 4:2:0, o que leva a taxas de compressão mais elevadas.

### 1. Codec híbrido sem perdas

#### 1) Parâmetro *search\_area*

Neste ponto variou-se o parâmetro da área de pesquisa dos blocos nas frames inter.

Utilizando os parâmetros fixos indicados, obteve-se a seguinte tabela:

- *estimation*: adaptative
- *estimationBlockSize* = 100
- *initial\_m* = 10
- *predictor* = LINEAR\_JPEG\_7
- *search\_mode* = INTERSPERSED
- *block\_size* = 8
- *intraFramePeriodicity* = 10

**Tabela 1 - Taxa de compressão em função de *search\_area***

Vídeo	<i>search_area</i>	Taxa de compressão	Tempo de compressão (segundos)	Tempo de descompressão (segundos)
ducks_take_off	16	2.83	25	5.1
	32	2.83	66	4.9
	64	2.83	217	5.2
in_to_tree	16	3.16	26	5.3
	32	3.17	70	5.2
	64	3.16	220	5.2
old_town	16	3.22	28	6.6
	32	3.22	70	6.6
	64	-	-	-

Da tabela retira-se que o valor com melhor relação entre a taxa de compressão e o tempo de compressão é o valor 16.

### 2) Parâmetro *block\_size*

Neste ponto variou-se o parâmetro do tamanho dos blocos nas frames inter.

Utilizando os parâmetros fixos indicados, obteve-se a seguinte tabela:

- *estimation*: adaptative
- *estimationBlockSize* = 100
- *initial\_m* = 10
- *predictor* = LINEAR\_JPEG\_7
- *search\_mode* = INTERSPERSED
- *search\_area* = 16
- *intraFramePeriodicity* = 10

**Tabela 2 - Taxa de compressão em função de *block\_size***

Vídeo	<i>block_size</i>	Taxa de compressão	Tempo de compressão (segundos)	Tempo de descompressão (segundos)
ducks_take_off	2	2.13	54	8.3
	4	2.71	28	5.8
	8	2.77	26	5.1
in_to_tree	2	2.26	54	8.4
	4	3.0	28	5.6
	8	3.1	26	5.3
old_town	2	2.26	66	8.9
	4	3.05	32	6.7
	8	3.22	28	6.8

Da tabela retira-se que o valor com melhor relação entre a taxa de compressão e o tempo de compressão é o valor 8.

### 3) Parâmetro *intraFramePeriodicity*

Neste ponto variou-se o parâmetro da periodicidade entre frames intra.

Utilizando os parâmetros fixos indicados, obteve-se a seguinte tabela:

- *estimation*: adaptative
- *estimationBlockSize* = 100
- *initial\_m* = 10
- *predictor* = LINEAR\_JPEG\_7
- *search\_mode* = INTERSPERSED
- *search\_area* = 16
- *block\_size* = 8

**Tabela 3 - Taxa de compressão em função de *intraFramePeriodicity***

Vídeo	<i>intraFramePeriodicity</i>	Taxa de compressão	Tempo de compressão (segundos)	Tempo de descompressão (segundos)
ducks_take_off	5	2.82	23	5.1
	10	2.83	25	5.1
	15	2.77	26	5.1
	20	2.77	26	5.1
in_to_tree	5	3.17	24	6.0
	10	3.16	26	5.3
	15	3.08	26	5.3
	20	3.08	27	5.1
old_town	5	3.22	26	6.6
	10	3.22	28	6.8
	15	3.22	28	6.8
	20	3.22	29	6.7

Da tabela retira-se que o valor com melhor relação entre a taxa de compressão e o tempo de compressão é o valor 5.

#### 4) Parâmetro *estimationBlockSize*

Neste ponto variou-se o parâmetro do número de blocos a utilizar na estimação dinâmica do parâmetro M do codificador de Golomb.

Utilizando os parâmetros fixos indicados, obteve-se a seguinte tabela:

- *estimation*: adaptative
- *initial\_m* = 10
- *predictor* = LINEAR\_JPEG\_7
- *search\_mode* = INTERSPERSED
- *search\_area* = 16
- *block\_size* = 8
- *intraFramePeriodicity* = 5

**Tabela 4 - Taxa de compressão em função de *estimationBlockSize***

Vídeo	<i>estimationBlockSize</i>	Taxa de compressão	Tempo de compressão (segundos)	Tempo de descompressão (segundos)
ducks_take_off	100	2.82	23	5.1
	200	2.83	23	5.1
	500	2.83	23	5.1
	1000	2.82	23	4.9
in_to_tree	100	3.17	24	6.0
	200	3.17	23	5.1
	500	3.17	24	5.1
	1000	3.16	24	5.1
old_town	100	3.22	25	6.5
	200	3.22	25	6.5
	500	3.22	26	6.7
	1000	3.22	25	6.7

Da tabela retira-se que não existe nenhum valor testado com melhor relação entre a taxa de compressão e o tempo de compressão, pelo que se escolheu o valor 100.

#### 5) Parâmetro *search\_mode*

Após realizar alguns dos testes anteriores com a pesquisa no modo exaustivo, verificou-se que a taxa de compressão aumenta ligeiramente (aprox. 6%) com o custo do tempo de compressão aumentar significativamente (cerca de 11 vezes maior). Assim, optou-se por escolher o modo de pesquisa INTERSPERSED com os valores escolhidos anteriormente.

## 2. Codec híbrido com perdas

Neste ponto variou-se o número de bits perdidos em cada componente de cor (Y, U e V).

Calculou-se o PSNR entre o vídeo original e o vídeo descomprimido com perdas utilizando a ferramenta *ffmpeg*.

Uma vez que os vídeos original e descomprimido não têm o mesmo formato de cor (4:4:4 e 4:2:0), apenas faz sentido analisar o valor do PSNR para a componente de cor Y.

Utilizando os parâmetros fixos indicados, obteve-se a seguinte tabela:

- *estimation*: adaptative
- *estimationBlockSize* = 100
- *initial\_m* = 10
- *predictor* = LINEAR\_JPEG\_7
- *search\_mode* = INTERSPERSED
- *search\_area* = 16
- *block\_size* = 8
- *intraFramePeriodicity* = 5

**Tabela 5 - Taxa de compressão em função do número de bits perdidos**

Vídeo	Nº de bits perdidos nas componentes Y, U e V	PSNR (Y)	Taxa de compressão	Tempo de compressão (segundos)	Tempo de descompressão (segundos)
ducks_take_off	1	51.14	3.05	22.7	4.95
	2	42.69	3.22	22.6	4.85
	3	35.70	3.33	23.4	4.60
	4	29.26	3.30	22.7	4.63
	5	12.64	3.05	22.3	3.83
in_to_tree	1	51.14	3.46	22.2	4.76
	2	42.70	3.66	22.1	4.55
	3	35.72	3.67	21.9	4.34
	4	29.53	3.57	22.1	4.25
	5	11.52	3.60	22.4	3.60
old_town	1	51.15	3.53	24.5	5.87
	2	42.71	3.70	24.4	5.57
	3	35.74	3.74	24.0	5.57
	4	29.61	3.71	23.7	5.28
	5	13.09	3.72	23.0	4.42

Da tabela retira-se que o valor com melhor relação entre a taxa de compressão e o PSNR é o valor 2, uma vez que com 3 ou mais bits perdidos o PSNR diminui bastante e a taxa de compressão diminui apenas ligeiramente.

### III. TESTES FINAIS

Nesta secção foram realizados testes com os quatro vídeos recomendados, utilizando os parâmetros escolhidos na secção anterior. Nas tabelas abaixo encontram-se as taxas de compressão, os tempos de execução e o PSNR, quando aplicável.

Os parâmetros utilizados nos testes foram os seguintes:

- *estimation*: adaptative
- *estimationBlockSize* = 100
- *initial\_m* = 10
- *predictor* = LINEAR\_JPEG\_7
- *search\_mode* = INTERSPERSED
- *search\_area* = 16
- *block\_size* = 8
- *intraFramePeriodicity* = 5

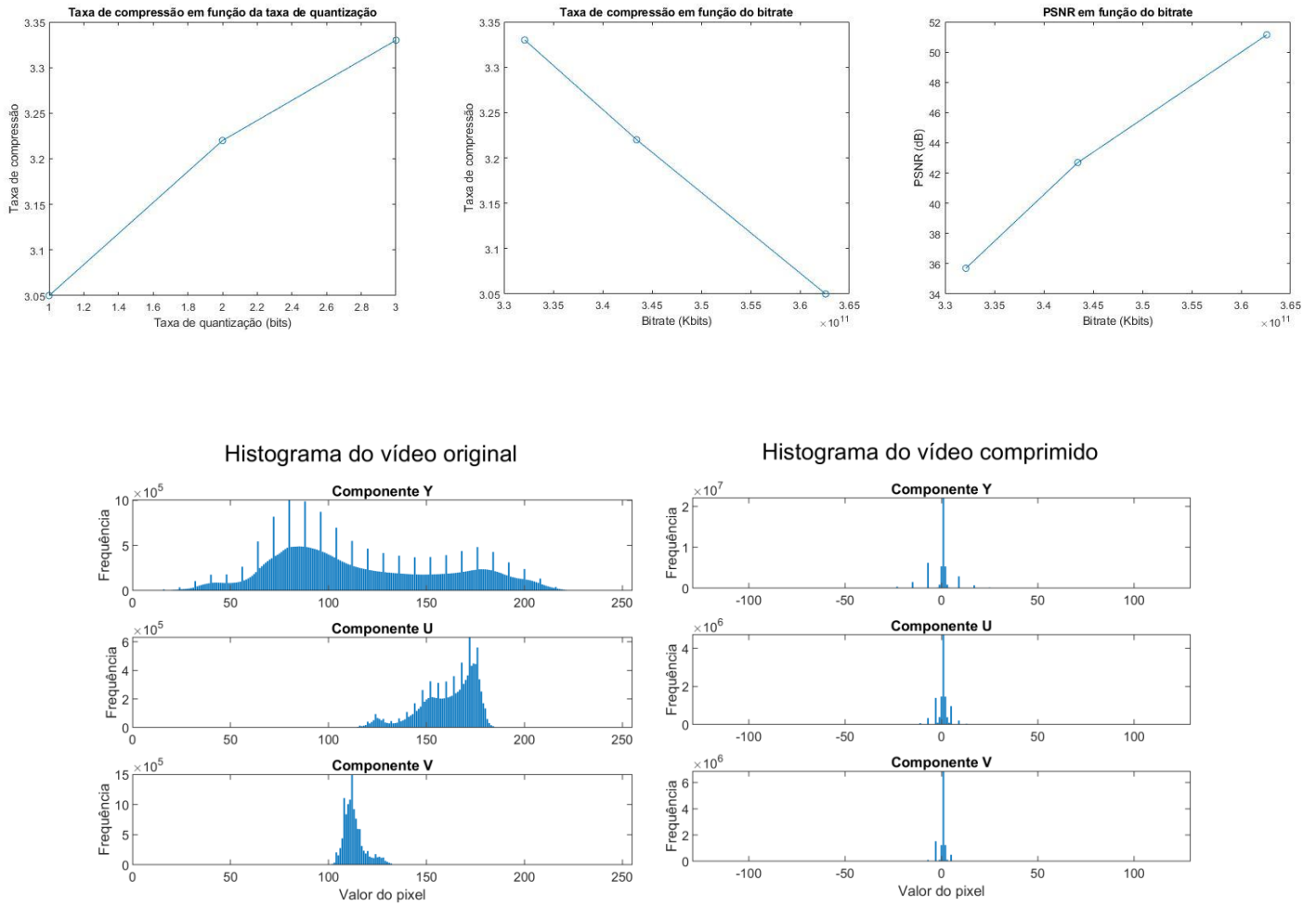
**Tabela 6 - Taxa de compressão sem perdas**

Vídeo	Taxa de compressão	Tempo de compressão (segundos)	Tempo de descompressão (segundos)
ducks_take_off	2.70	259	45
in_to_tree	2.99	239	45
old_town	3.21	251	67
park_joy	2.56	255	47

**Tabela 7 - Taxa de compressão com perdas**

Vídeo	Taxa de compressão	PSNR (Y)	Tempo de compressão (segundos)	Tempo de descompressão (segundos)
ducks_take_off	3.07	42.69	253	66
in_to_tree	3.45	42.69	223	44
old_town	3.67	42.71	246	56
park_joy	2.89	42.69	226	42

Apresentam-se agora alguns gráficos relevantes, traçados com os dados obtidos, bem como um histograma de um dos vídeos original e comprimido.



#### IV. CONTRIBUIÇÕES

Foi abordado o desempenho de cada um dos elementos do grupo e achou-se que o Gonçalo Cardoso deveria ter menos 1 valor que os colegas, sendo considerado justo que estes fossem distribuídos pelos restantes elementos, visto que as suas tarefas foram desempenhadas com sucesso e estes não deveriam ser prejudicados.