Codificação de Áudio e Vídeo

Luis Assunção, Joel Pinheiro
DETI, Universidade de Aveiro
Aveiro, Portugal
{pedroassuncao, joelpinheiro}@ua.pt

***Resumo* – Este terceiro trabalho teve como objetivo desenvolver programas que permite alterar a resolução de um ficheiro de video codificado no formato RGB, calcular PSNR entre dois videos, compressão baseado num algoritmo Intra-Frame e outro em Inter-Frame.**

***Palavras chave* – Intra-Frame, Inter-Frame, Codec, JPEG, JPEG-LS, Codificador, Descodificador, PSNR, Recise**

I. INTRODUÇÃO

Neste terceiro trabalho, tinhamos como tarefa desenvolver 4 modulos de software:

- **A**lterar a resolução de um video
- **C**alcular o PSNR entre 2 videos
- **C**odec de vídeo Intra-Frame
- **C**odec de vídeo Inter-Frame

Internamente ao processo de codificaçao, temos como requisito a utilizaçao do espço de cores YUV/Y CbCr 4:2:0. Caso o vídeo recebido nao se encontre neste espaço de cores, sera necessário converter o mesmo, o que podería implicar erros relativamente ao vídeo original. Este projeto esta dividido em quatro partes. A primeira parte tem como objetivo ambientar-nos com os conceitos necessarios para o desenvolvimento do codificador de vídeo, enquanto que as tres fases seguintes consistem no desenvolvimento de diferentes versoes do codec. Apos a preparacção do ambiente para o desenvolvimento deste trabalho, foram utilizadas as classes utilizados do trabalho anterior (BitStream e Golomb).

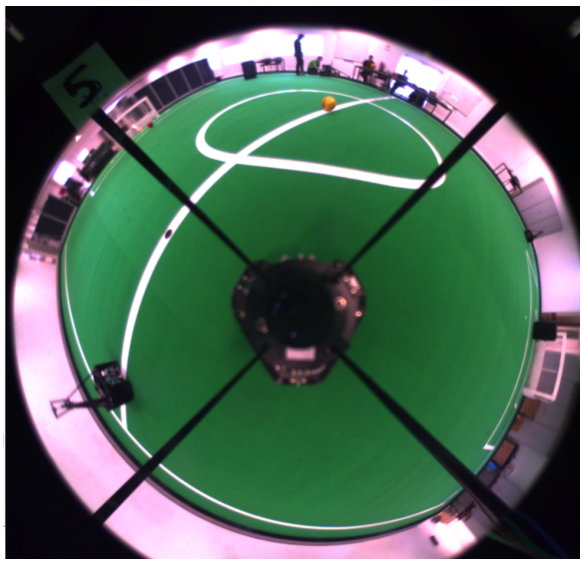


Figura 1: cabada1.rgb

$$X_p = \begin{cases} min(a,b), se \geq max(a,b) \\ max(a,b), se \leq min(a,b) \\ a+b-c, casocontrario \end{cases}$$

Figura 2: JPEG-LS

II. CODIFICAÇÃO LOSSLESS

Num codificador Intraframe, o vídeo e codificado frame a frame de forma consecutiva e independente. No contexto deste trabalho, tínhamos a possibilidade de optar por o previsor linear JPEG ou pelo previsor nao linear JPEG-LS; no nosso caso optamos por utilizar os dois.

Neste previsor de tipo espacial, para um determinado pixel, sao tidos em conta os valores de pixeis vizinhos para calcular a previsao do seu valor. Considerando que pretendemos determinar o valor da previsao do pixel x que se encontra na posiçãao representada abaixo. Importa referir que este pixel nao se encontra em qualquer extremidade da frame.

A previsao do valor do pixel é determinada com base na deteção de bordas, isto é, se existir uma borda horizontal (acima de x), vai ser utilizado para a previsao o valor do pixel a, enquanto que se existir uma borda vertical (a esquerda de x) vai ser utilizado para a previsao o valor do pixel b. Caso nao exista nenhuma destas bordas, vão ser utilizados para a previsao os valores dos pixeis a, b e c.

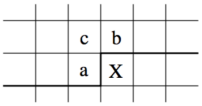
Por fim, para determinar o valor a ser utilizado pelo codificador de entropia, tem de ser obtido o valor do erro, isto e, o valor de r. O valor de r é dado pela subtração entre o valor original do pixel e a previsao do seu valor.

III. CODIFICAÇÃO LOSSY

IV. CLASSES IMPLEMENTADAS ESTRUTURA

Na elaboração deste trabalho prático, foram implementadas várias classes, mediante os requisitos da implementação.

Listing 1: RawIntraFrameCodecLossless.h



Mode	Predictor
1	a
2	b
3	c
4	$a+b-c$
5	$a+(b-c)/2$
6	$b+(a-c)/2$
7	$(a+b)/2$

Figura 3: JPEG 7 Linear Prediction Modes

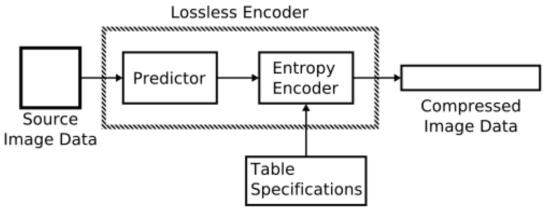


Figura 4: Codificador sem perdas

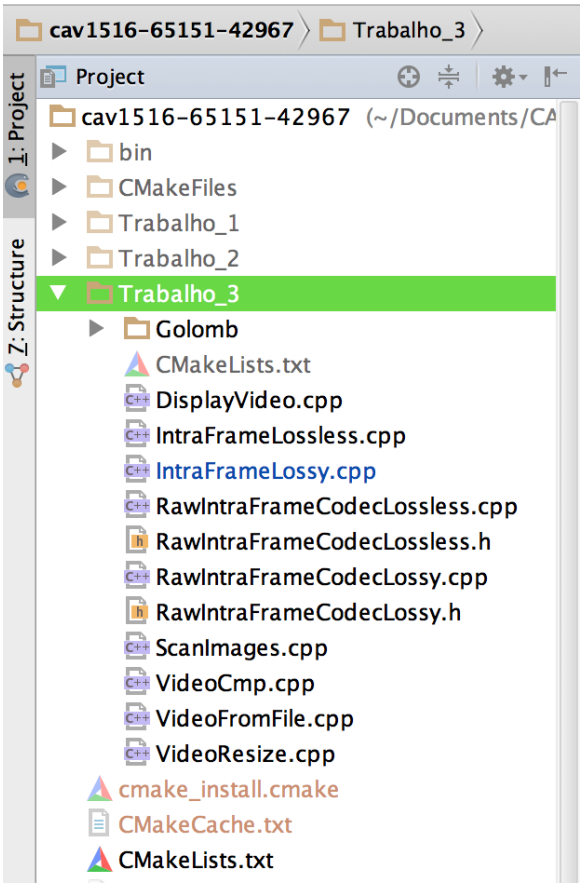


Figura 5: Estrutura de código

```
#ifndef CAV1516_RAWINTRAFRAMECODEC_H
#define CAV1516_RAWINTRAFRAMECODEC_H
#include "opencv2/opencv.hpp"

using namespace cv;

class RawIntraFrameCodecLossless {
public:
    RawIntraFrameCodecLossless(char*
        inputFilePath, char* outputFilePath);
    void intraFrameEncode(int predictorMode, int
        showVideo, int m);
    void intraFrameDecode(int predictorMode, int
        showVideo);

protected:
private:
    Mat* img;
    Mat* intermediate;

    Mat* resize;

    Mat* predY;
    Mat* predU;
    Mat* predV;

    Mat* yyMat;

    Mat* uuMat;
    Mat* vvMat;

    char* inputFilePath;
    char* outputFilepath;
    Mat* resizeFrame(Mat* matToResize);
    void predictMat(int predictorMode);
    short predictValue(int predictorMode, Mat
        *xxMat, int i, int j);
};

#endif //CAV1516_RAWINTRAFRAMEENCODER_H
```

Listing 2: RawIntraFrameCodecLossy.h

```
//
// Created by root on 12/2/15.
//

#ifndef CAV1516_RAWINTRAFRAMECODEC_H
#define CAV1516_RAWINTRAFRAMECODEC_H
#include "opencv2/opencv.hpp"

using namespace cv;

class RawIntraFrameCodecLossy {
public:
    RawIntraFrameCodecLossy(char* inputFilePath,
        char* outputFilePath);
    void intraFrameEncode(int predictorMode, int
        q_y, int q_cr, int q_cb, int showVideo,
        int m);
    void intraFrameDecode(int predictorMode, int
        showVideo);

protected:
private:
    Mat* img;
    Mat* intermediate;

    Mat* resize;

    Mat* predY;
    Mat* predU;
    Mat* predV;

    Mat* yyMat;

    Mat* uuMat;
    Mat* vvMat;

    char* inputFilePath;
    char* outputFilepath;
    Mat* resizeFrame(Mat* matToResize);
```

```
    void predictMat(int predictorMode);
    short predictValue(int predictorMode, Mat
        *xxMat, int i, int j);
};

#endif //CAV1516_RAWINTRAFRAMEENCODER_H
```

V. MANUAL DE UTILIZAÇÃO

Nesta secção iremos exemplificar como se poderá executar o nosso projecto:

Para compilar o nosso projecto foi desenvolvida uma Cmakelist. Apenas é necessário executar dois comandos:

Listing 3: Cmake Make

```
$ cmake cmakelists.ttx
$ make
```

Este exercício apenas é necessário correr para que seja mostrada o video em tempo real da WebCam.

Listing 4: Exercício 2: DisplayVideo

```
$ ./DisplayVideo
```

Neste exercício é necessário passar como argumento de execução do programa o caminho dos dois ficheiros de vídeo que queremos comparar. No final é retornado o calculo do valor de PSN obtido para cada um dos canais de vídeo.

Listing 5: Exercício 4. VideoFromFile

```
$ ./VideoFromFile
    /Users/joelpinheiro/Downloads/cambada1.rgb

Usage: <File Path>
```

Este programa serve para redimensionar um vídeo.

Listing 6: Executar VideoResize

```
$ ./VideoResize ~/Downloads/cambada2.rgb
    ~/Downloads/x.rgb 50 50

Usage: referenceVideo resizedVIdeo verticalRes
    horizontalRes
```

Este programa fazer encode/decode lossless intraframe. Os modos de compressão vão de 1 a 8. De 1 a 7 são os 7 modos de predição linear JPEG, o 8º é o JPEG-LS.

Listing 7: Executar IntraFrameLossless

```
$ ./IntraFrameLossless
    /Users/joelpinheiro/Downloads/cambada1.rgb
    /Users/joelpinheiro/Downloads/cambada2.rgb
    /Users/joelpinheiro/Downloads/cambada3.rgb 8
    1 1

Usage: referenceVideo compressedVIdeo
    decompressedVideo compressMode
    imageShow(0-f, 1-v) m
```

Este programa fazer encode/decode lossy intraframe. Os modos de compressão vão de 1 a 8. De 1 a 7 são os 7 modos de predição linear JPEG, o 8º é o JPEG-LS.

Tabela I: Lossless referente ao primeiro ficheiro RGB

Resultados codificação Lossless			
m	Original (B)	Compressed (B)	Modo de Comp
1	207618065	50233404	1
1	207618065	41164138	8
8	207618065	61674312	1
8	207618065	60456434	8

Tabela II: Lossy referente ao primeiro ficheiro RGB

Resultados codificação Lossy					
m	q_y	q_cr	q_cb	Original (B)	Compressed (B)
1	1	2	3	207618065	41204537
1	4	8	8	207618065	17652051
1	1	2	2	207618065	33908788
1	4	8	8	207618065	15246322
4	1	2	3	207618065	48660083

Listing 8: Executar IntraFrameLossy

```
$ ./IntraFrameLossy
    /Users/joelpinheiro/Downloads/cambada1.rgb
    /Users/joelpinheiro/Downloads/cambada2.rgb
    /Users/joelpinheiro/Downloads/cambada3.rgb 8
    2 2 2 1 1

Usage: referenceVideo compressedVIdeo
    decompressedVideo compressMode y_quantizer
    cr_quantizer cb_quantizer imageShow(0-f,
    1-v) M
```

VI. ANÁLISE DOS RESULTADOS EXPERIMENTAIS

.

VII. NOTAS FINAIS

Relativamente aos nossos algoritmos desenvolvidos, e as tecnicas utilizadas, podemos concluir que comparando com o JPEG linear, o JPEG-LS é mais eficiente e fornece imagens de maior qualidade produzindo entropias mais baixas no calculo dos residuais, traduzindo-se em taxas de compressão superiores. Os algoritmos Lossy permitem atingit taxas de compressão mais elevadas, no entanto com grande perda de qualidade de imagem.

REFERÊNCIAS

[1] https://en.wikipedia.org/wiki/Lossless_$JPEG$