

INSTITUTO SUPERIOR DE  
ENGENHARIA DE LISBOA  
ADEETC

LICENCIATURA EM ENGENHARIA INFORMÁTICA E  
MULTIMÉDIA

SEMESTRE DE VERÃO

4º TRABALHO PRÁTICO

---

**Codificação de Sinais Multimédia**

---

*Realizado por:*

João SANTOS nº 39348

Rui SANTOS nº 39286

Junho, 2017



# Conteúdo

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introdução</b>  | <b>3</b>  |
| <b>2</b> | <b>Desenvolvimento</b>                                     | <b>4</b>  |
| 2.1      | Codificação considerando cada frame como uma intra-frame . | 4         |
| 2.2      | Codificação inter-frame sem compensação de Movimento . . . | 5         |
| 2.3      | Codificação inter-frame com compensação de movimento . . . | 8         |
| 2.4      | Pesquisa full-search . . . . .                             | 9         |
| 2.5      | Predição de movimento . . . . .                            | 10        |
| 2.6      | Codificação . . . . .                                      | 11        |
| 2.7      | Descodificação . . . . .                                   | 12        |
| 2.8      | Resultados obtidos . . . . .                               | 12        |
| <b>3</b> | <b>Conclusões</b>  | <b>13</b> |

## Lista de Figuras

|    |  |    |
|----|--|----|
| 1  | Frame 1 codificada com uma qualidade de 10%. . . . . | 4  |
| 2  | Frame 1 codificada com uma qualidade de 75%. . . . . | 4  |
| 3  | Taxas de Compressão. . . . .                         | 5  |
| 4  | Relação Sinal-Ruído . . . . .                        | 5  |
| 5  | Entropia. . . . .                                    | 5  |
| 6  | Energia média por píxel. . . . .                     | 5  |
| 7  | Intra-frame (I). . . . .                             | 6  |
| 8  | Diferença entre frames. . . . .                      | 6  |
| 9  | Descodificação da frame 8. . . . .                   | 6  |
| 10 | Taxas de Compressão. . . . .                         | 7  |
| 11 | Relação Sinal-Ruído. . . . .                         | 7  |
| 12 | Entropia. . . . .                                    | 7  |
| 13 | Energia média p/ píxel. . . . .                      | 7  |
| 14 | Algoritmo de pesquisa full-search. . . . .           | 9  |
| 15 | Predição de movimento. . . . .                       | 10 |
| 16 | Frame 2 codificada. . . . .                          | 11 |
| 17 | Vetores de movimento. . . . .                        | 11 |
| 18 | Frame 2 descodificada. . . . .                       | 12 |
| 19 | Taxas de Compressão. . . . .                         | 13 |
| 20 | Relação Sinal-Ruído. . . . .                         | 13 |
| 21 | Entropia. . . . .                                    | 13 |
| 22 | Energia média p/ píxel. . . . .                      | 13 |

# 1 Introdução

O quarto trabalho prático da disciplina de Codificação de Sinais Multimédia tem como objetivo o estudo e a implementação dos princípios básicos da codificação de vídeo. Através de um conjunto de frames disponibilizados, vão ser implementadas três formas distintas de codificação de vídeo, das quais:

1. Codificação de cada frame da sequência de vídeo utilizando um codificador JPEG, como o desenvolvido no trabalho 3, considerando que cada frame é uma intra-frame (I).
2. No segundo ponto é considerado que todas as frames são inter-frames (P) à exceção da primeira que é uma intra-frame (I). As inter-frames (P) são obtidas através da diferença entre a frame a codificar e a intra-frame (I). O ponto 2 é implementado **sem** compensação de movimento.
3. Tal como no ponto 2, é considerado que as frames são inter-frames (P) à exceção da primeira que é uma intra-frame (I). Neste ponto do trabalho é realizada a predição da frame a codificar com base na I-frame **com** compensação de movimento.

Com os três tipos de codificação referidos em cima implementados foram desenvolvidos métodos que permitem calcular os valores da Taxa de Compressão, relação sinal-ruído, entropia da frame a transmitir, energia média por píxel e o tempo de compressão e descompressão de cada ponto referido em cima. Posteriormente vão ser apresentados os resultados obtidos em forma de tabelas e gráficos para cada um dos resultados.

## 2 Desenvolvimento

### 2.1 Codificação considerando cada frame como uma intra-frame

O primeiro método de codificação desenvolvido no trabalho requer uma codificação considerando cada frame como uma intra-frame (I). Neste caso, cada frame da sequência de vídeo é codificada sem qualquer relação com as frames anteriores. Para tal, foi utilizado o codificador JPEG disponibilizado na biblioteca do OpenCV. Ou seja, para cada frame da sequência de vídeo disponibilizada foi aplicado o seguinte método:

```
cv2.imwrite("frame_x.jpeg",img_x,(cv2.IMWRITE_JPEG_QUALITY,qualidade))
```

Esta função recebe como argumentos o *pathname* de cada frame, o *array* da frame em questão e realiza a codificação baseada num fator de qualidade introduzido por nós. É possível observar nas Figuras 1 e 2 a codificação realizada para dois fatores de qualidade distintos, respetivamente, 10 e 75%.



Figura 1: Frame 1 codificada com uma qualidade de 10%.



Figura 2: Frame 1 codificada com uma qualidade de 75%.

Seguidamente foram realizado os cálculos das Taxas de Compressão, Relação Sinal-Ruído, Entropia e Energia média por píxel. Sendo os resultados obtidos apresentados nas Figuras 3, 4, 5 e 6, para motivos de simplificação os dados apresentados nos gráficos são referentes apenas a um fator de qualidade de 75%.

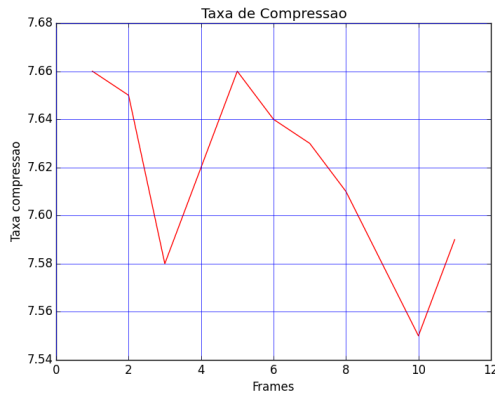


Figura 3: Taxas de Compressão.

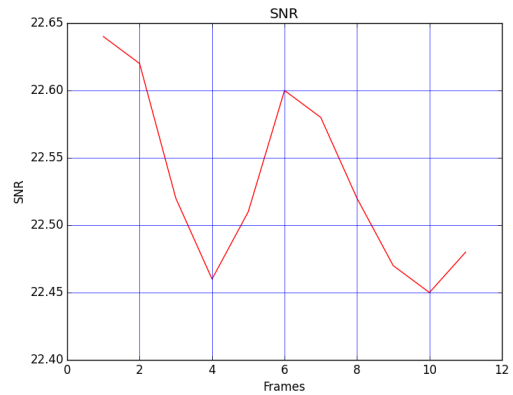


Figura 4: Relação Sinal-Ruído

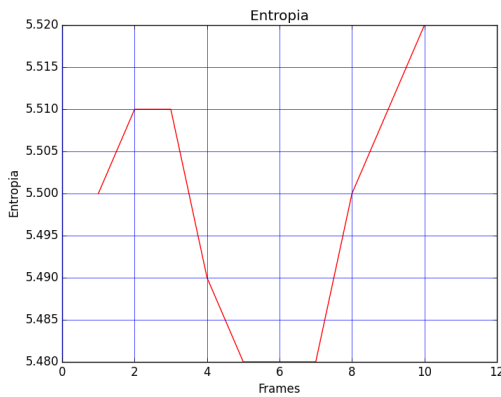


Figura 5: Entropia.

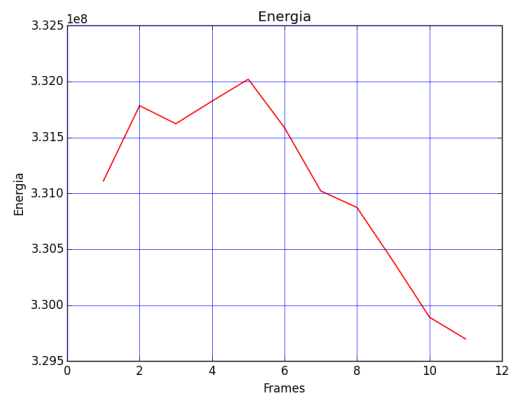


Figura 6: Energia média por píxel.

## 2.2 Codificação inter-frame sem compensação de Movimento

O segundo método de codificação, tem como objetivo, a codificação das frames de vídeo considerando estas como inter-frames (P), à exceção da primeira, que é uma intra-frame (I). Neste modo, é realizada a codificação da intra-frame (I) do mesmo modo que se realizou no capítulo anterior, de seguida, cada frame (P) é codificada realizando a diferença entre a intra-frame (I) e a P frame atual, finalmente é realizada a codificação da diferença, sem compensação de movimento. No que diz respeito ao decodificador, o processo realizado é o inverso do codificador.

É possível observar nas Figuras 7, 8 e 9 os resultados obtidos na codificação e na decodificação.



Figura 7: Intra-frame (I).

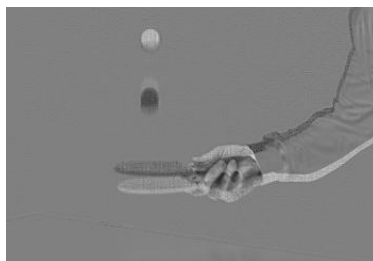


Figura 8: Diferença entre frames.



Figura 9: Descodificação da frame 8.

A Figura 7 representa a intra-frame (I) à qual foi realizada uma codificação JPEG. A Figura 8 representa a codificação da diferença entre a frame (I) e a frame 8 da sequência de vídeo. Comparando a Figura 7 e 8 é possível observar na Figura 8 que todo o fundo da imagem permaneceu igual, pois não existiu movimento, e apenas a zona da bola e da raquete é que foi codificada. Na Figura 9 temos um exemplo da descodificação da frame 8.

Tal como na parte 1 do trabalho foram realizados os cálculos das Taxas de Compressão, Relação Sinal-Ruído, Entropia e Energia média por píxel. É possível observar nas Figuras 10, 11, 12 e 13 os resultados obtidos em comparação com o codificador implementado na parte 1 do trabalho.

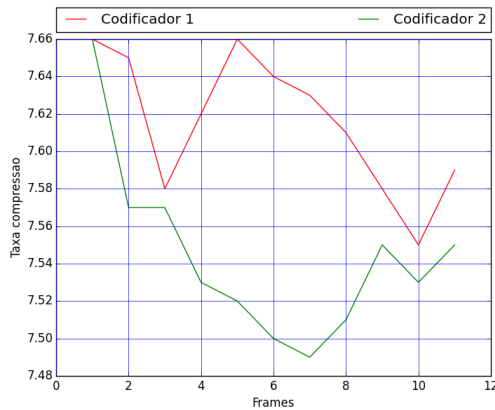


Figura 10: Taxas de Compressão.

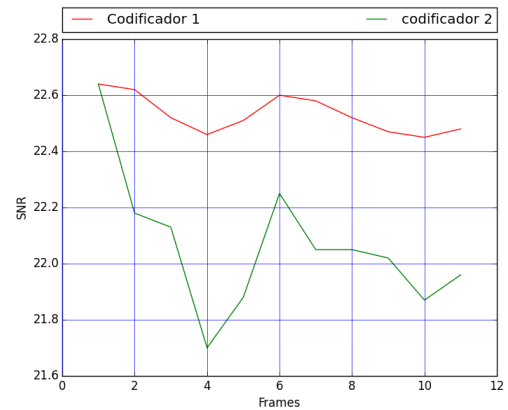


Figura 11: Relação Sinal-Ruído.

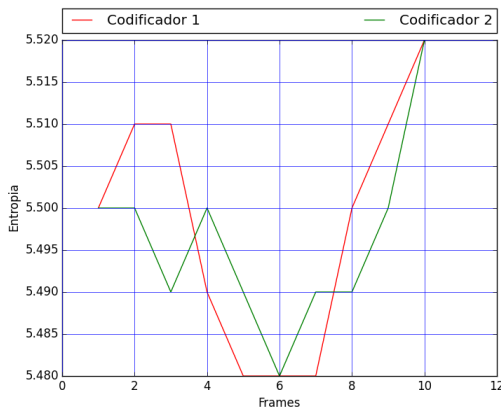


Figura 12: Entropia.

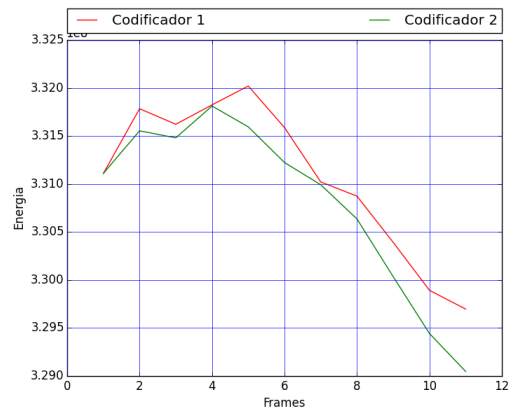


Figura 13: Energia média p/ píxel.

Observando os resultados obtidos podemos reparar que na Figura 10 existe uma diminuição na Taxa de Compressão no codificador dois, isto significa que são necessários um maior número de bits para representar as imagens utilizando o codificador do segundo ponto do trabalho.

No que diz respeito à Relação Sinal-Ruído apresentada na Figura 11, é possível observar uma ligeira descida da mesma, deste modo, implica uma maior introdução de ruído por parte do codificador 2.

Tanto a energia média por píxel como a Entropia mantiveram-se significativamente idênticas à exceção de umas ligeiras oscilações.



## 2.3 Codificação inter-frame com compensação de movimento

O terceiro ponto do trabalho prático requer a implementação de uma codificação inter-frame utilizando compensação de movimento. É considerado que cada frame à exceção da primeira é uma inter-frame (P). Ao contrário do ponto anterior a predição da frame a codificar é implementada com base na I-frame utilizando a compensação de movimento. Finalmente, a frame a transmitir é a diferença entre a frame a codificar e a sua predição.

De modo a implementar este codificador foi necessário construir as seguintes funções:

- Uma função que permita realizar a medição do erro absoluto médio (MAE) entre dois blocos;
- Uma função que realize a pesquisa do bloco da frame a codificar numa janela de pesquisa (-15 a +15) da frame I-frame;
- Uma função que percorra os blocos da frame a codificar e construa a frame predita;

## 2.4 Pesquisa full-search

O tipo de pesquisa implementada no codificador do trabalho foi a pesquisa full-search, para realizar este tipo de pesquisa começamos por selecionar um bloco de 16x16 na janela de procura e medido o erro absoluto médio entre esse bloco e o bloco atual recebido. Caso o valor obtido seja menor que o que foi dados inicialmente como referência, significa que o bloco que estamos a analisar é o mais parecido ao bloco atual. O bloco mais aproximado é então guardado e o valor do erro mínimo absoluto é atualizado.

Após a pesquisa estar concluída obtemos o bloco mais aproximado ao ao bloco P e as coordenadas desse bloco.

É possível observar na Figura 14 o código desenvolvido que permite a implementação da procura full-search.

```
def full_search(bloco_16x16, janela_pesquisa, janela_x, janela_y, Pframex, Pframey):

    altura, largura = janela_pesquisa.shape
    minimo = 10**20
    array = 0
    coluna = 0
    linha = 0

    for i in np.arange(altura-15):
        for j in np.arange(largura-15):

            bloco = janela_pesquisa[i:i+16, j:j+16]

            # Função que permite realizar o calculo do erro absoluto médio
            # entre ambos os blocos
            diff = MAE(bloco_16x16,bloco)

            if (diff < minimo):
                coluna = i
                linha = j
                array = bloco
                minimo = diff

    x = janela_x + coluna
    y = janela_y + linha

    coordx = x - Pframex
    coordy = y - Pframey

    return array, coordx, coordy
```

Figura 14: Algoritmo de pesquisa full-search.

## 2.5 Predição de movimento

De forma a percorrer os blocos da frame a codificar, são selecionados os blocos de 16x16 de forma progressiva na imagem recebida e selecionada também uma janela de pesquisa com dimensão -15 a 15. Após selecionadas ambas as janelas é realizada a pesquisa full-search apresentada na Figura 14.

De modo a selecionar a janela de pesquisa é calculado o valor máximo que o  $x$  e o  $y$  podem tomar. Seguidamente, são realizadas as exceções caso o bloco esteja a ocupar a primeira coluna, a primeira linha, a ultima coluna e a ultima linha.

Podemos observar na Figura 15 o algoritmo desenvolvido que permite realizar a predição de movimento.

```
def predicao_movimento(i_frame, p_frame):

    idx = 0
    altura , largura = i_frame.shape
    array_final = np.zeros(altura * largura)
    array_final = array_final.reshape(altura, largura)

    vetores_mov = np.zeros(((altura*largura)/(16*16))*2)
    vetores_mov = vetores_mov.reshape((altura*largura)/(16*16),2)

    for col in np.arange(0,altura,16):
        for lin in np.arange(0,largura,16):

            # Janela de pesquisa [-15, 15]
            ini = [col - 15, lin - 15]
            fim = [(col + 16) + 15, (lin + 16) + 15]

            # Criar bolo 16x16
            blocoP_16x16 = p_frame[col:col+16, lin:lin+16]

            if (ini[0] < 0):
                ini[0] = 0
            if (ini[1] < 0):
                ini[1] = 0
            if (fim[0] > altura):
                fim[0] = altura
            if (fim[1] > largura):
                fim[1] = largura

            janela_pesquisa = i_frame[ini[0]:fim[0],ini[1]:fim[1]]

            bloco, coluna, linha = full_search(blocoP_16x16, janela_pesquisa, ini[0], ini[1], col, lin)
            vetores_mov[idx] = coluna , linha
            array_final[col:col+16,lin:lin+16] = bloco
            idx += 1

    codificado = p_frame.astype(float) - array_final.astype(float) + 128.
    return codificado , vetores_mov
```

Figura 15: Predição de movimento.

## 2.6 Codificação

Para a implementação do codificador é necessário fazer a predição de movimento e o método de pesquisa full-search para todas as imagens. O codificador implementado devolve os vetores de movimento que vão ser utilizados posteriormente para realizar a descodificação das frames. Com o codificador desenvolvido é possível observar na Figura 16 a frame 2 resultante da codificação onde é possível observar no "fundo" a frame (I) utilizada para a predição de movimento.

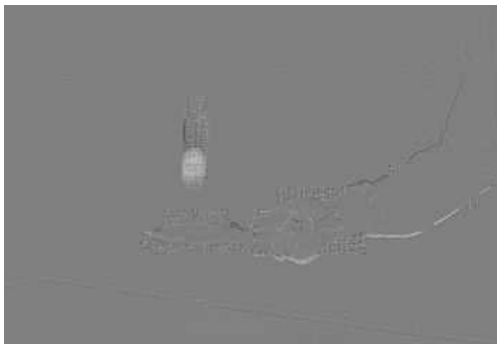


Figura 16: Frame 2 codificada.

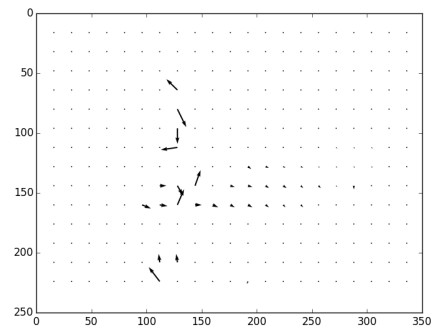


Figura 17: Vetores de movimento.

## 2.7 Descodificação

Para efetuar a decodificação das imagens e obter as frames originais, foi criada uma função que recebe como argumentos os vetores retornados pela codificação. De seguida, é obtida a frame de referência (I-frame) e são percorridas as outras imagens, onde vamos criar uma matriz de zeros com o tamanho da frame atual e selecionamos o primeiro bloco de 16x16 fazendo-o corresponder a um bloco de 16x16 da frame de referência. Para efetuar essa correspondência utilizamos os vetores de movimento. Caso os vetores de movimento não sejam os dois nulos o bloco desloca-se, caso contrário o bloco é o primeiro e fazemos a correspondência entre os mesmos.

No final a matriz de zeros irá ser totalmente preenchida por blocos da frame de referência e é finalizada a decodificação realizando a soma entre ela e a frame com perda de movimento e é obtida a imagem original.

É possível observar na Figura 18 a frame 2 decodificada, que resulta do processo de decodificação da Figura 16.



Figura 18: Frame 2 decodificada.

## 2.8 Resultados obtidos

Com a decodificação de todas as frames realizada foram efetuados os cálculos das Taxas de Compressão, Relação Sinal-Ruído, entropia e energia de cada uma. É possível observar nas Figuras seguintes a comparação entre os resultados obtidos nos codificadores 1 e 2 em comparação com o codificador implementado neste ponto.

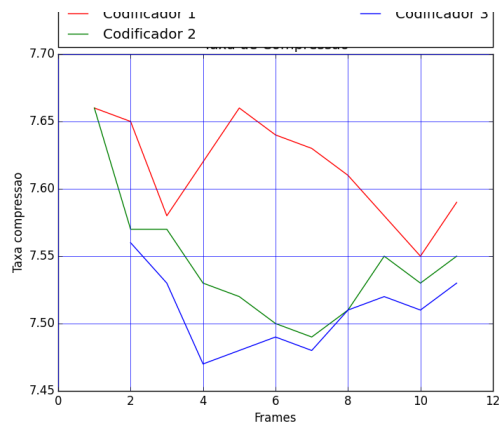


Figura 19: Taxas de Compressão.

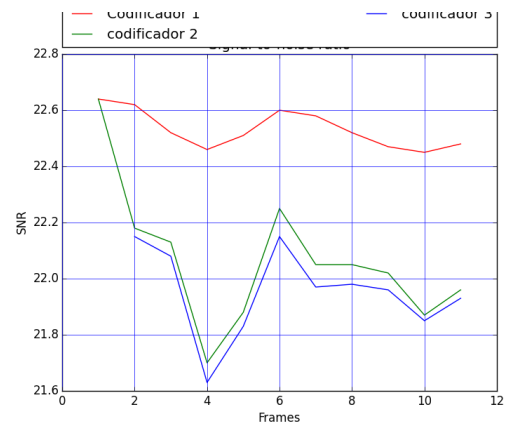


Figura 20: Relação Sinal-Ruído.

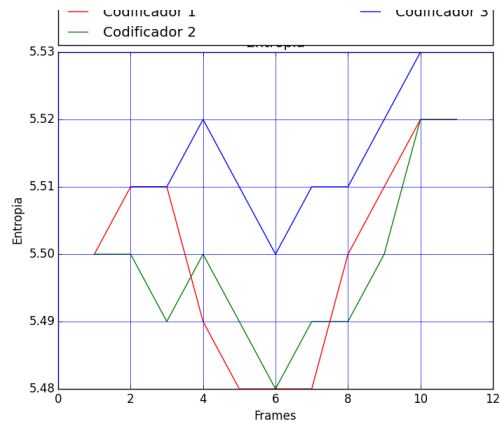


Figura 21: Entropia.

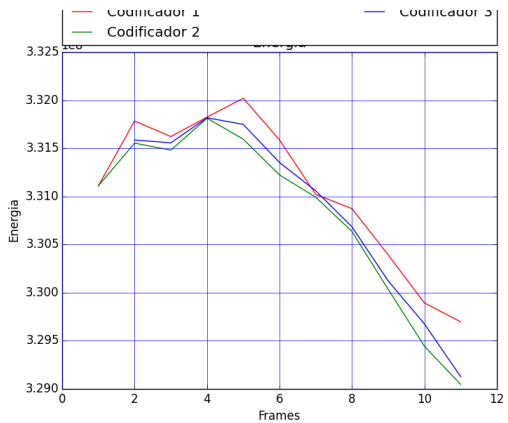


Figura 22: Energia média p/ pixel.

É possível observar através dos gráficos apresentados que existiu uma ligeira descida nos valores das Taxas de Compressão em comparação com os codificadores anteriores. A relação Sinal-Ruído manteve-se muito aproximada ao codificador 2, embora tenha existido um aumento na Entropia. Já os valores da energia média por pixel mantiveram-se muito aproximados dos anteriores.

### 3 Conclusões

Observando os resultados obtidos podemos concluir que o trabalho final da disciplina de Codificação de Sinais Multimédia foi desenvolvido com sucesso. Foram implementados todos os pontos existentes no enunciado e os resultados obtidos vão de acordo com o esperado. As imagens resultados da descodificação em todos os pontos são muito aproximadas das imagens originais e é possível observar isso através dos gráficos obtidos. Apenas o codificador do terceiro ponto do trabalho apresenta um tempo de codificação muito elevado pois o processo de codificação com estimação de movimento requer muito processamento. Em suma o grupo encontra-se satisfeito com o desenvolvimento do trabalho.