**Practica de Codificación de fuente**



**Parte 1 – compresión de imagen – Octave/Matlab**

**Entregar un scrip de Matlab/Python que realice las siguientes tareas:**

1. Abrir la imagen lena.tiff, convertir la imagen a niveles de gris, y tres imagenes con únicamente las componente de Rojo, verde y azul respectivamente
2. amplificar la componente de rojo unicamente a travez de una variable
3. Realizar un histograma de los niveles de grises de la imagen.
4. aplicar la DCT a bloques de 8x8 y aplicar distintas matrices de cuantización y observar la variación de la calidad de la imagen.

**Parte 2**

**Entregar un scrip de Matlab/Python que realice las siguientes tareas:**

a) Restar los cuadros adyacentes de Minions.gif contenidos en el Minions.zip y estudiar el histograma del cuadro resultante.

Notas: tener en cuenta   
<https://la.mathworks.com/help/matlab/ref/imread.html>

<https://la.mathworks.com/help/images/ref/dct2.html>

<https://www.mathworks.com/examples/image/mw/images-ex44409888-image-compression-with-the-discrete-cosine-transform>

<https://la.mathworks.com/help/images/ref/imhist.html>

Abrir con elecardStreameye el archivo.ts

1. Que formato de compresión de video tiene y que niveles y perfiles

* Formato de compresión: H.264
* Perfil High
* Nivel 3.0

1. Que formato de codificación de crominancia/luminancia usa

* 4:2:0

1. Como está compuesto el GOP

1 Bloque I

5 grupos de 3 bloques B seguidos de un bloque P

3 bloques B

1. tomar el primer GOP y analizar los PTS-DTS de cada cuadro y explicar el orden de codificación y decodificación



|  |  |
| --- | --- |
| **TimeStamp** | **Acción** |
| 00:18:34:968 | Decodifico I |
| 00:18:35:008 | Decodifico B1 |
| 00:18:35:048 | Presento B2 |
| 00:18:35:088 | Presento B1 y Decodifico B3 |
| 00:18:35:128 | Presento B3 y Decodifico P1 |
| 00:18:35:168 | Presento I y Decodifico B4 |
| 00:18:35:208 | Presento B5 |
| 00:18:35:248 | Presento B4 y Decodifico B6 |
| 00:18:35:288 | Presento B6 decodifico P2 |
| 00:18:35:328 | Presento P1 y Decodifico B7 |
| 00:18:35:368 | Presento B8 |
| 00:18:35:408 | Presento B7 y Decodifico B9 |
| 00:18:35:448 | Presento B9 y Decodifico P3 |
| 00:18:35:488 | Presento P2 y Decodifico B10 |
| 00:18:35:528 | Presento B11 |
| 00:18:35:568 | Presento B10 y Decodifico B12 |
| 00:18:35:608 | Presento B12 y Decodifico P4 |
| 00:18:35:648 | Presento P3 y Decodifico B13 |
| 00:18:35:688 | Presento B14 |
| 00:18:35:728 | Presento B13 y Decodifico B15 |
| 00:18:35:768 | Presento B15 y Decodifico P5 |
| 00:18:35:808 | Presento P4 y Decodifico B16 |
| 00:18:35:848 | Presento B17 |
| 00:18:35:888 | Presento B16 y Decodifico B18 |
| 00:18:35:928 | Presento B18 y Decodifico I2 |
| 00:18:35:968 | Presento P5 y Decodifico B19 |

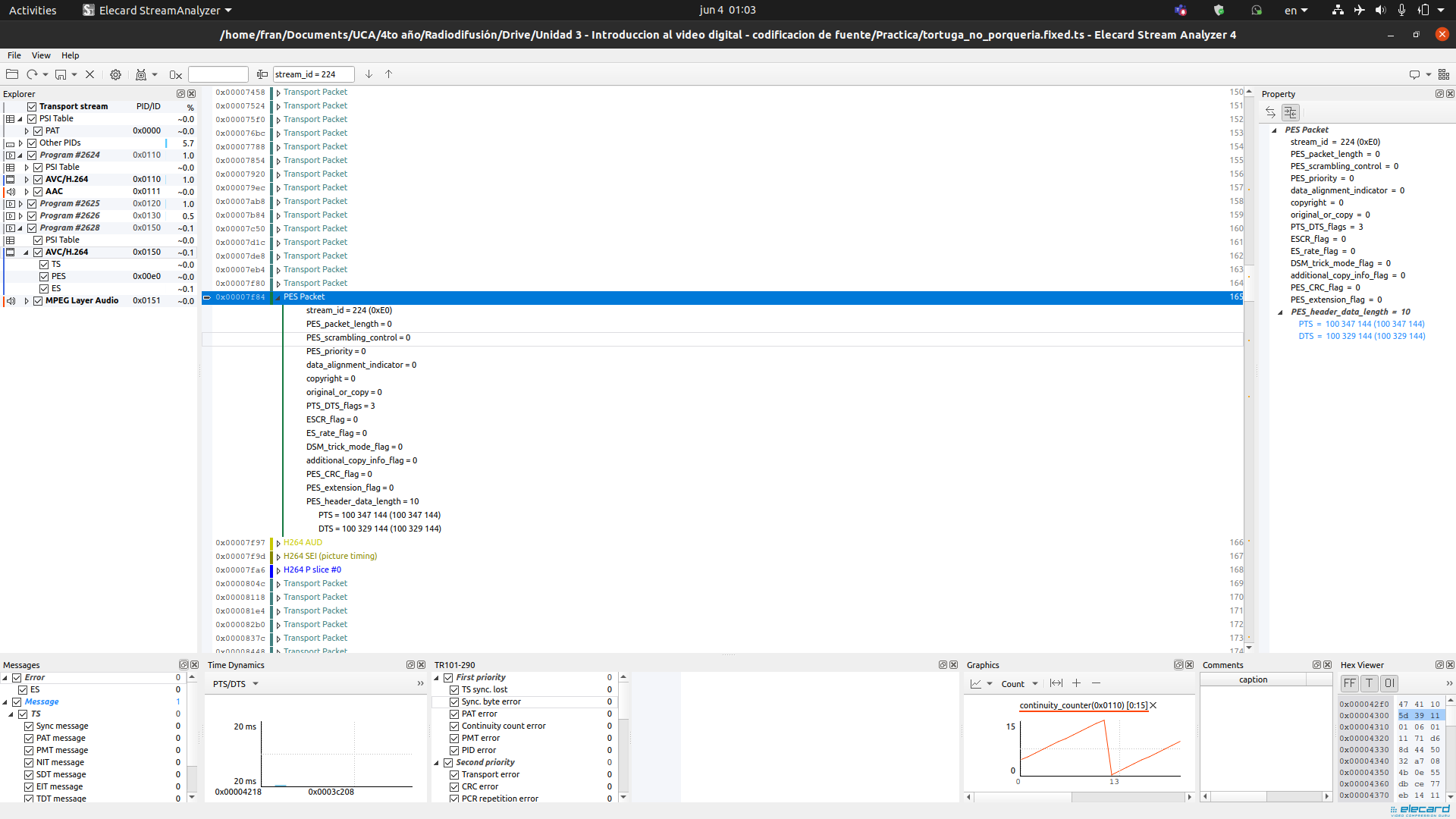
**Parte 3 - Capa de sistema MPEG**

Abrir con elecardStreamAnalyzer el archivo.ts

identificar todos los programas con sus PID y Construir la tabla PAT y PMT



Explicar la cabecera de un PES



La cabecera de un PES contiene obligatoriamente

* Payload\_unit\_start\_indicator(3 bytes) indica que ese PES es el comienzo
* stream\_id (1 byte) que indica ids de audio o video dependiendo a que corresponda el PES.
* PES\_packet\_length (2 bytes) que indica la cantidad de bytes que quedan en el paquete luego de ese campo. Puede ser cero, y cuando es cero significa que el paquete puede tener cualquier longitud. Solo se permite usar PES\_packet\_length = 0 cuando el payload sea un elementary stream de video.

Otros datos opcionales del header incluyen:

* data\_alignment\_indicator (1 bit) cuando está en 1 significa que el PES packet está automaticamente seguido del *video start code*
* PTS\_DTS\_flags (2 bits) indica si el paquete trae PTS y DTS.
  + 0 significa que no trae ninguna de las dos
  + 1 no está permitido
  + 2 el paquete solo trae PTS
  + 3 el paquete trae ambas PTS y DTS

identificar el PCR de un programa

En el transport packet con PID = 0x301 se recibe el PCR del programa 2426.

Se puede observar que este paquete tiene un PCR\_flag levantado y la información del PCR en el Adaptation field.

