**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 imágenes con únicamente la componente de Rojo, verde y azul respectivamente
2. amplificar la componente de rojo únicamente a través de una variable y reconstruir la imagen
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

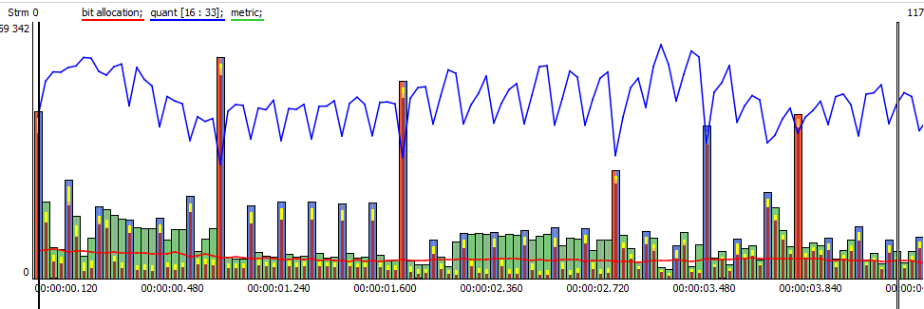


1. Que formato de codificación de crominancia/luminancia usa

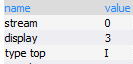


1. Como está compuesto el GOP

I – B – B – B – P – B – B – B – P – B – B – B – P - B – B – B – P – B – B – B – P – I – …



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

I:  B1: 

B2:  B3: 

P:  B4: 

B5:  B6: 

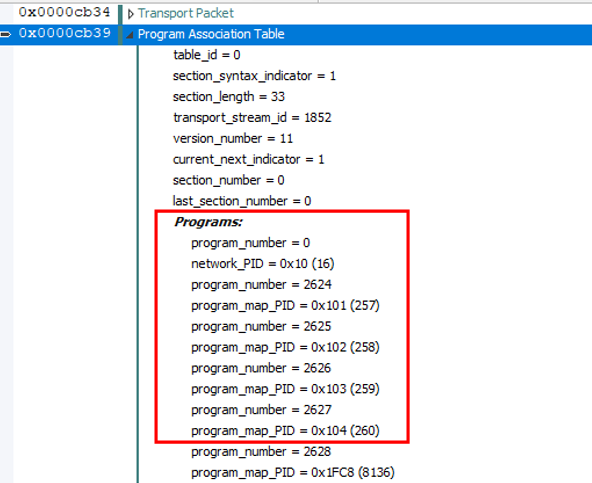
Orden de codificación: B2 – B1 – B3 – I – B5 – B4 – B6 – P

Orden de decodificación: I – B1 – B2 – B3 – P – B4 – B5 – B6

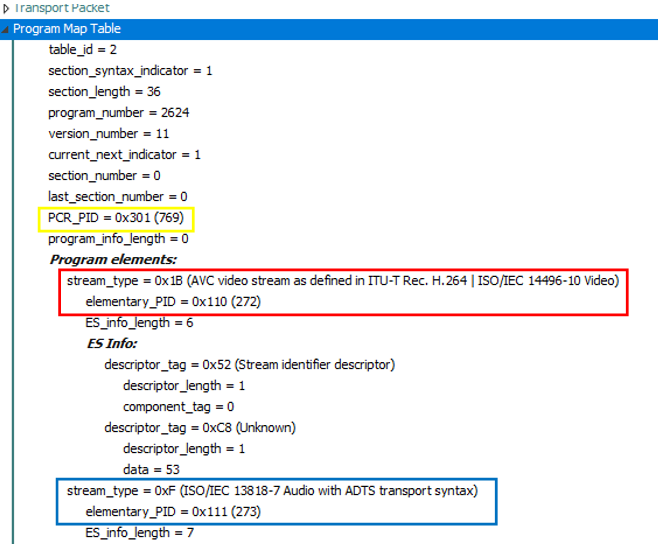
**Parte 3 - Capa de sistema MPEG**

Abrir con elecardStreamAnalyzer el archivo.ts

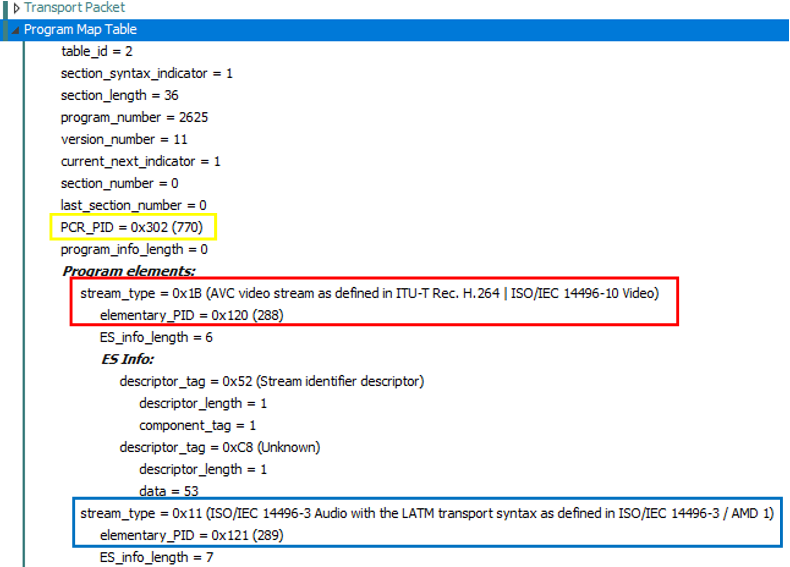
identificar todos los programas con sus PID y Construir la tabla PAT y PMT



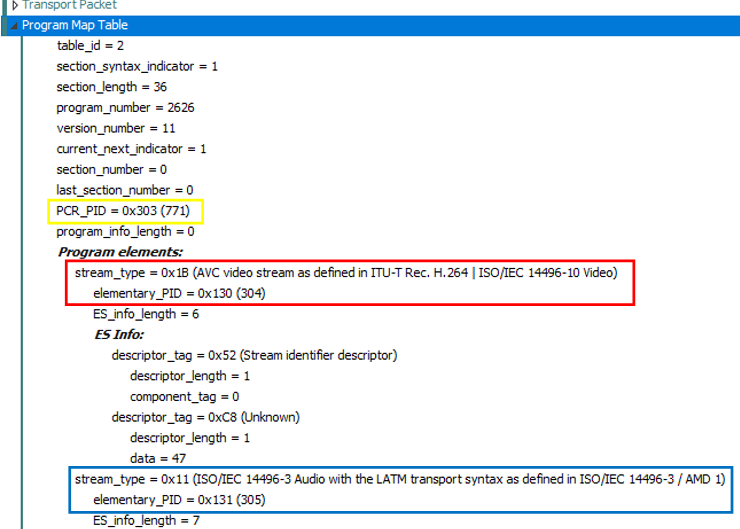
|  |  |
| --- | --- |
| **PAT** | |
| **Programa** | **PID** |
| 2624 | 0x101 (257) |
| 2625 | 0x102 (258) |
| 2626 | 0x103 (259) |
| 2627 | 0x104 (260) |



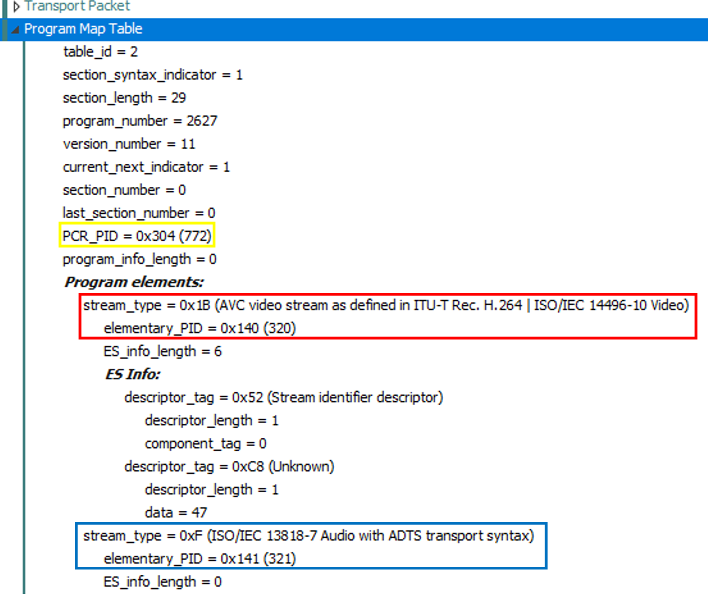
|  |  |
| --- | --- |
| **PMT (2624)** | |
| **PES** | **PID** |
| VIDEO | 0x110 (272) |
| AUDIO | 0x111 (273) |



|  |  |
| --- | --- |
| **PMT (2625)** | |
| **PES** | **PID** |
| VIDEO | 0x120 (288) |
| AUDIO | 0x121 (289) |

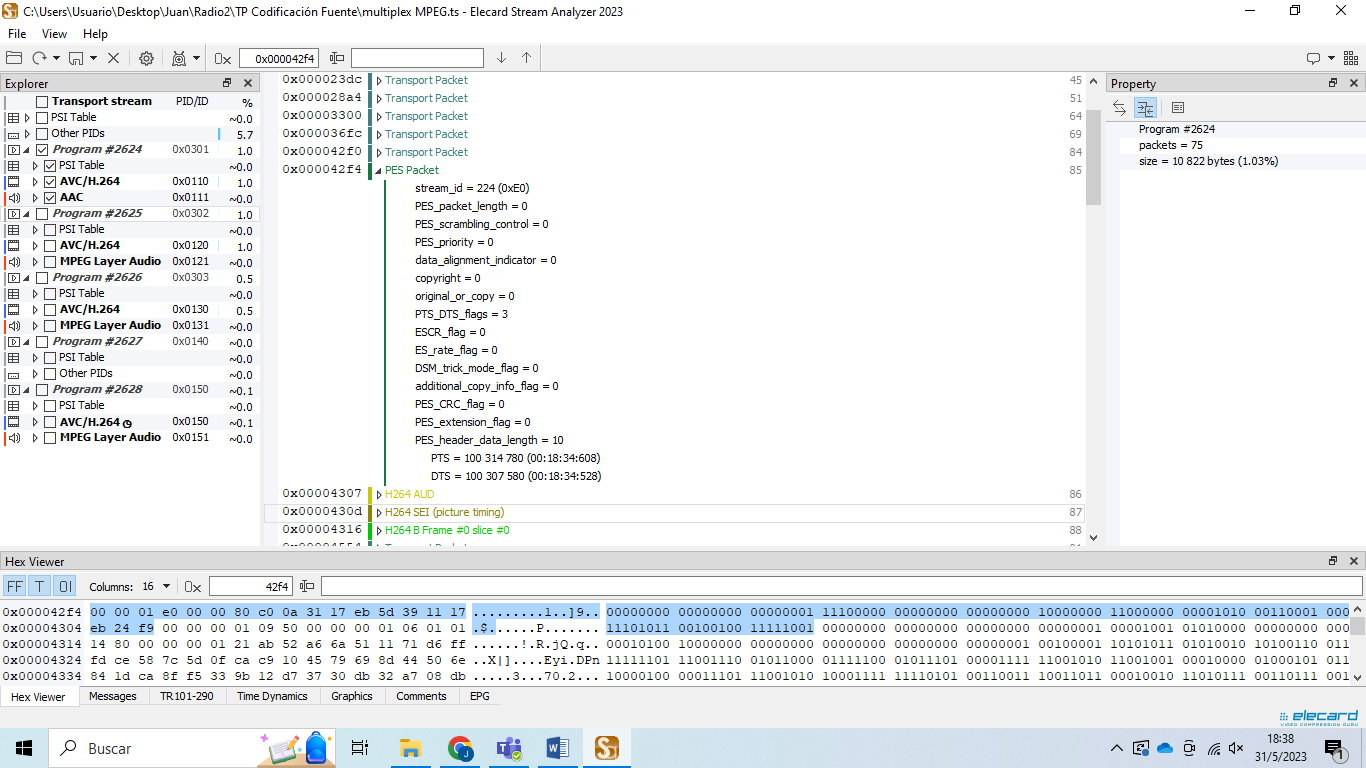


|  |  |
| --- | --- |
| **PMT (2626)** | |
| **PES** | **PID** |
| VIDEO | 0x130 (304) |
| AUDIO | 0x131 (305) |



|  |  |
| --- | --- |
| **PMT (2627)** | |
| **PES** | **PID** |
| VIDEO | 0x140 (320) |
| AUDIO | 0x141 (321) |

Explicar la cabecera de un PES



**stream\_id:** indica el tipo de flujo de datos contenido en el PES. Como es 0xE0 significa que se trata de video.

**PES\_packet\_length:** indica la longitud total del paquete, así el decodificador sabe cuántos bytes debe leer para procesar el paquete correctamente. Como es 0 significa que el tamaño del paquete es desconocido.

**PES\_scrambling\_control:** indica si los datos fueron encriptados. Como es 0 significa que los datos no están encriptados.

**PES\_priority:** indica la prioridad del PES en relación con otros paquetes. Como es 0 significa que no tiene una prioridad especial.

**data\_alignment\_indicator:** indica si los datos están alineados en límites de palabras. Como es 0 significa que no lo están.

**copyright:** indica si los datos están protegidos por derechos de autor. Como es 0 significa que no lo están.

**original\_or\_copy:** indica si los datos son originales. Como es 0 significa que son copias.

**PTS\_DTS\_flags:** indica la presencia de campos PTS y DTS en el paquete. Como es 3 significa que ambos están presentes en el paquete.

**ESCR\_flag:** indica la presencia del campo ESCR en el paquete. Como es 0 significa que no está.

**ES\_rate\_flag:** indica la presencia del campo ES Rate en el paquete. Como es 0 significa que no está presente.

**DSM\_trick\_mode\_flag:** indica si se está utilizando un modo de truco DSM en el paquete. Como es 0 significa que no.

**additional\_copy\_info\_flag:** indica la presencia de información adicional de copia en el paquete. Como es 0 significa que no hay.

**PES\_CRC\_flag:** indica si se ha incluido un campo CRC en el paquete para verificar la integridad de los datos. Como es 0 significa que no.

**PES\_extension\_flag:** indica la presencia de una extensión de PES en el paquete. Como es 0 significa que no la hay.

**PES\_header\_data\_length:** indica la longitud de los datos adicionales en la cabecera del PES. Como es 10 significa que hay 10 bytes reservados en la cabecera para almacenar datos adicionales.

Identificar el PCR de un programa

