TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA INFORMÁTICA

CURSO 2015-2016

INTEGRIDAD EN VÍDEOS DE DISPOSITIVOS MÓVILES

Roumen Daton Medenou

Directores:

Luis Javier García Villalba

Ana Lucila Sandoval Orozco

Departamento de Ingeniería del Software e Inteligencia Artificial

---

FACULTAD DE INFORMÁTICA

UNIVERSIDAD COMPLUTENSE DE MADRID

El presente Trabajo Fin de Grado se enmarca dentro de un proyecto de investigación titulado RAMSES aprobado por la Comisión Europea dentro del Programa Marco de Investigación e Innovación Horizonte 2020 y en el que participa el Grupo GASS del Departamento de Ingeniería del Software e Inteligencia Artificial de la Facultad de Informática de la Universidad Complutense de Madrid (Grupo de Análisis, Seguridad y Sistemas, http://gass.ucm.es, grupo 910623 del catálogo de grupos de investigación reconocidos por la UCM).

Por razones de confidencialidad del proyecto se ha omitido información del trabajo desarrollado para no infringir la normativa correspondiente.

_____

Roumen Daton Medenou

Agradecimientos

Quiero agradecer a mis directores la posibilidad de realizar el presente trabajo, agradecimientos que extiendo al resto de miembros del Grupo GASS, por haberme enseñado y ayudado tanto.

Resumen

Con frecuencia una persona conectada a Internet sube vídeos a una red social desde su móvil. Esto es así por la popularización de los smartphones de bajo coste y por el incremento del uso de las redes sociales multimedia. Cuando se visionan éstos, surge la pregunta: ¿Habrá sido editado? ¿Es verdadero? Actualmente, las técnicas forenses para imágenes son bastante sofisticadas. En cambio, las orientadas a vídeos digitales y, particularmente, a vídeos digitales de dispositivos móviles, están apenas en sus comienzos. Este trabajo forma parte de un proyecto de investigación financiado por la Comisión Europea dentro del Programa Marco de Investigación e Innovación Horizonte 2020 denominado RAMSES que tiene, entre otros objetivos, el desarrollo de una potente herramienta de análisis forense para vídeos digitales de dispositivos móviles. Más concretamente, la contribución que aporta el presente trabajo a la citada herramienta es el diseño e implementación de un método de análisis forense que indica al usuario las zonas del vídeo que son poco fiables, lo cual es muy útil a la hora de detectar manipulaciones.

Palabras clave

Análisis forense, dispositivo móvil, herramienta, integridad, manipulación, redes sociales, Theia, vídeos.

Abstract

People, connected to the Internet, on a regular basis, upload videos to Social Networks, using mobile phones. This is due to the popularisation of low-end Smartphones, and to the rise of Multimedia Oriented Social Networks. When we watch a homemade video, we ask ourselves if it has been edited, whether it's fake or not. When working with photographs, forensic techniques and methods are quite sophisticated. On the other side, implementations of forensic methods applied to Digital Video, are still in an early stage. This work is a part of a much bigger project, financed by The EU Framework Programme for Research and Innovation Horizon 2020, named as RAMSES. It has objectives such as developing a powerful Forensic Tool on mobile devices digital video, among others. More specifically, the contribution made by this work to the above mentioned tool, is the design and implementation of a forensic analysis technique which pinpoints the not very reliable areas of a video. That is quite useful when detecting tampering.

Keywords

Forensic analysis, integrity, manipulation, mobile device, social network, Theia, tool, videos.

.

Lista de Acrónimos

AAC	Advanced Audio Coding
AVC	Advanced Vídeo Coding
B-Frames	Bi-predictive Frame
CFA	Color Filter Array
CLUT	Colour Lookup Table
CPU	Central Processing Unit
DCT	Discrete Cosine Transform
DQ	Double Quantification
DSC	Digital Still Camera
GOP	Group of Pictures
GPS	Global Position System
I-Frames	Intra-coded Frame
I-MB	Intra MacroBlock
ITU	International Telecommunication Union
JPEG	Join Photograph Expert Group
MOV	Quick Time Movie Format
MP4	Media Player 4
MPEG	Moving Picture Expert Group

PC	Personal Computer
P-Frames	Predicted frames
P-MB	Predicted MacroBlock
S-MB	Switching Macroblock
VPF	Variation Prediction Footprint
YUV	Intensity-Hue-value

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

1. INTRODUCCIÓN

1.1. Motivación

Los dispositivos móviles actuales, están permanentemente interconectados. En la era del 'Internet de las cosas', y los dispositivos móviles actuales no son exactamente ordenadores personales. Mediante demonios de mensajería, clientes a redes sociales, o incluso malware, los dispositivos móviles pueden, sin intervención alguna, recibir, enviar y almacenar piezas de información, ya sean imágenes, ficheros de video, etc.

Cuando una persona se encuentra, por sorpresa, con un fichero de video en su móvil se hace las siguientes preguntas: ¿lo he grabado yo? ¿es un video vanilla (no-manipulado)?

Los metadatos, a día de hoy, son algo opcional, y el software cliente de redes sociales suele borrarlos. Por lo tanto, es necesaria la implementación de técnicas de análisis forense de vídeos, que permitan a las personas y autoridades poder generar ciertos datos que caractericen dichos videos.

1.2. Contexto

El presente Trabajo Fin de Grado se enmarca dentro de un proyecto de investigación titulado RAMSES aprobado por la Comisión Europea dentro del Programa Marco de Investigación e Innovación Horizonte 2020 (Convocatoria H2020-FCT-2015, Acción de Innovación, Número de Propuesta: 700326)y en el que participa el Grupo GASS del Departamento de Ingeniería del Software e Inteligencia Artificial de la Facultad de Informática de la Universidad Complutense de Madrid (Grupo de Análisis, Seguridad y Sistemas, http://gass.ucm.es, grupo 910623 del catálogo de grupos de investigación reconocidos por la UCM).

Además de la Universidad Complutense de Madrid participan las siguientes entidades:

- Treelogic Telemática y Lógica Racional para la Empresa Europea SL (España)
- Ministério da Justiça (Portugal)
- University of Kent (Reino Unido)
- Centro Ricerche e Studi su Sicurezza e Criminalità (Italia)
- Fachhochschule fur Offentliche Verwaltung und Rechtspflege in Bayern (Alemania)
- Trilateral Research & Consulting LLP (Reino Unido)
- Politecnico di Milano (Italia)
- Service Public Federal Interieur (Bélgica)
- Universitaet des Saarlandes (Alemania)
- Dirección General de Policía - Ministerio del Interior (España)

1.3. Objetivos

El presente Trabajo Fin de Grado (TFG) tiene los siguientes objetivos:

- Estudiar métodos existentes de caracterización de vídeos digitales y su eficacia.
- Elegir uno o varios métodos para determinar la integridad de un video.
- Depurar y Documentar la instalación y uso del Framework de Desarrollo en GNU/Linux, a fin de prepararlo para multiplataforma (tanto multi-S.O como en varias arquitecturas hardware).

1.4. Plan de Trabajo

El proyecto está compuesto de cuatro fases: Fase Previa, Documentación, Diseño e Implementación. Asimismo, cada fase contiene sus actividades entre las cuales destacan: Diseño, Implementación y Pruebas. Todas éstas se han desarrollado en paralelo, pero según la fase en la que se encontraba el proyecto, cada una se efectuaba con mayor o menor intensidad.

La Figura 1.1 muestra el diagrama de Gantt del proyecto.

Nombre de tarea	Duración (días)	Inicio	Fin
Fase Previa	**92**	**03/10/15**	**07/02/16**
● Planificación	15		
● Estudio de trabajos previos	30		
● Estudio de tecnologías existentes	15		
● Despliegue y actualización de plataforma de desarrollo	32		
Fase de Documentación	**30**	**08/02/16**	**21/03/16**
● Planificación	20		
● Análisis de Riesgos	10		
Fase de Diseño	**30**	**22/03/16**	**28/03/16**
● Toma de requisitos	10		
● Especificación de Objetivos	200		
● Preparación de la memoria	15		
Fase de Implementación	**40**	**10/05/16**	**04/07/16**
Fase de Pruebas	**14**	**05/07/16**	**24/07/16**

Tabla 1.1. Actividades de la fase de ejecución del proyecto

Id	Nombre de tarea
1	Planificación
2	Estudio trabajos previos
3	Estudio de tecnologías existentes
4	Despliegue y actualización de Plataforma de Desarrollo
5	
6	Planificación
7	Análisis de Riesgos
8	
9	Toma de Requisitos
10	Especficación de Objetivos
11	Preparación de la Memoria
12	Fase de Implementación
13	Fase de Pruebas

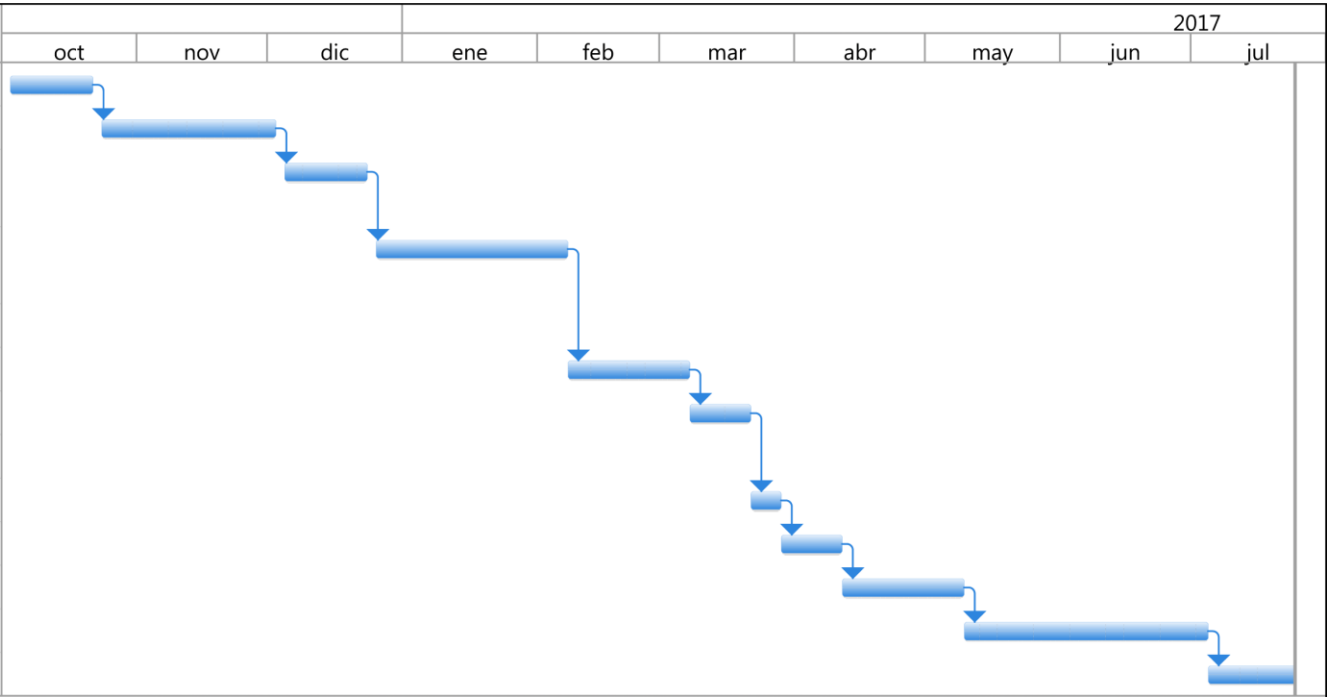


Figura 1.1: Diagrama de Gantt de la planificación del proyecto

1.5. Estructura de la memoria

El resto del trabajo está organizado en 7 capítulos con la siguiente estructura.

En el capítulo 2 se introducen conceptos básicos para comprender el análisis forense en vídeos de dispositivos móviles, esto es, el proceso de generación de un vídeo digital y los elementos de la cámara que sirven de base para las técnicas forenses en vídeos. También se realiza un estudio de la compresión y los tipos de codificación de un vídeo y una descripción de la metodología más utilizada para almacenar videos en Android, que es el H264 dentro de contenedor mp4.

En el capítulo 3 se presenta un estado del arte sobre técnicas forenses relacionadas con la detección de manipulaciones de vídeos de dispositivos móviles.

En el capítulo 4 se especifica una herramienta para el análisis forense de imágenes y vídeos digitales denominada *Theia,* describiendo sus principales características y funcionalidades.

En el capítulo 5 se presenta la contribución de este trabajo, que consiste en la implementación de un algoritmo de comprobación de la integridad de un vídeo con formato MP4. Asimismo, se presenta un experimento realizado para validar la eficacia del algoritmo.

En el Capítulo 6 se presentan las principales conclusiones extraídas de este trabajo y las líneas de trabajo futuro.

En los capítulos 7 y 8 se realiza un resumen en inglés de la introducción y las conclusiones del trabajo.

2. ANÁLISIS FORENSE EN VÍDEOS DIGITALES

Para comprender las diferentes técnicas de análisis forense de en vídeo digitales, primero se requiere conocer cómo está compuesta una cámara fotográfica y cuál es el procedimiento que se realiza para generar un vídeo.

Según [Moe12] a menudo se aplica el marco general que se ilustra en la Figura 2.1.

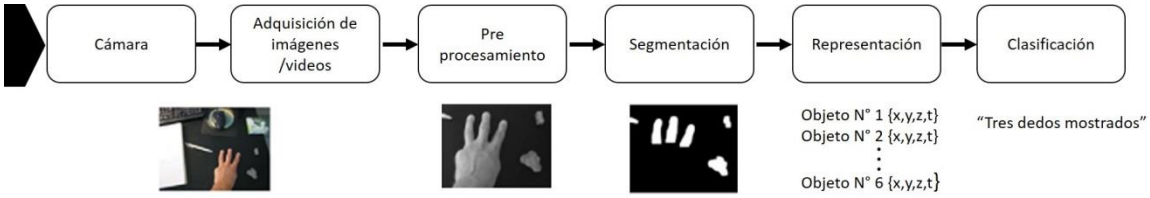


Figura 2.1 Marco general de un sistema de procesamiento de imágenes y vídeo

El sistema trabaja con vídeos e imágenes y no todos los bloques están incluidos en un sistema en particular, sin embargo el marco general proporciona una guía muy útil. A continuación se describe el propósito de los diferentes bloques:

- **Adquisición de imágenes y vídeos**: Se efectúa todo lo relacionado con la cámara y la configuración de su sistema. Por ejemplo, el tipo de cámara, ajustes de cámara, la óptica y la fuente de luz.

- **Pre procesamiento**: Se ejecutan ciertos procesos a la imagen antes de una transformación real. Por ejemplo convertir la imagen de color a escala de grises o recortar la parte más interesante de la imagen.

- **Segmentación**: Se extrae a información de interés de la imagen o datos de vídeo. En el ejemplo de la Figura 2.1. la información son los dedos. La imagen de la parte inferior muestra que se han segmentado los dedos incluido algo de ruido.

- **Representación**: Los objetos extraídos en la segmentación son representados de una manera concisa, por ejemplo el uso de algunos números representativos como se ilustra en la figura.

- **Clasificación**: Examina la información producida por el bloque de representación y clasifica el objeto como un objeto de interés o no. En el ejemplo se muestra objetos presentes de los dedos y por lo tanto la producción de este.

2.1. Proceso de Generación de un Vídeo

En el proceso de generación de un vídeo, conocido como *pipeline* [THN+] [Inc12], se combinan imagen, audio y datos. Su estructura es similar en dispositivos del mismo tipo, diferenciándose sólo en la calidad de la cámara o las prestaciones adicionales que ofrece.

Muchos de los detalles del proceso de generación de un vídeo en una cámara digital son propios de cada fabricante y tipo de dispositivo. Sin embargo, hay una estructura general que es común en todas las cámaras. Los componentes de una cámara digital Son: un sistema de lentes, un grupo de filtros, una matriz de filtro de colores o *Color Filter Array* (CFA), un sensor de imagen y un procesador de imagen o *Digital Image Processor* (DIP) [BSM08].

La generación del vídeo sigue un proceso similar al de generación de una imagen hasta el proceso realizado por el DIP que forma parte del procesador digital de señales o *Digital Signal Processor* (DSP), en el que se adiciona un micrófono y un conversor de señal analógica a digital. El proceso de generación de una imagen ha sido ampliamente descrito en la literatura.

Para generar un vídeo se ejecutan dos acciones en paralelo: El procesamiento de secuencia de imágenes y el procesamiento de audio. Este proceso se muestra esquemáticamente en la Figura 2.2.

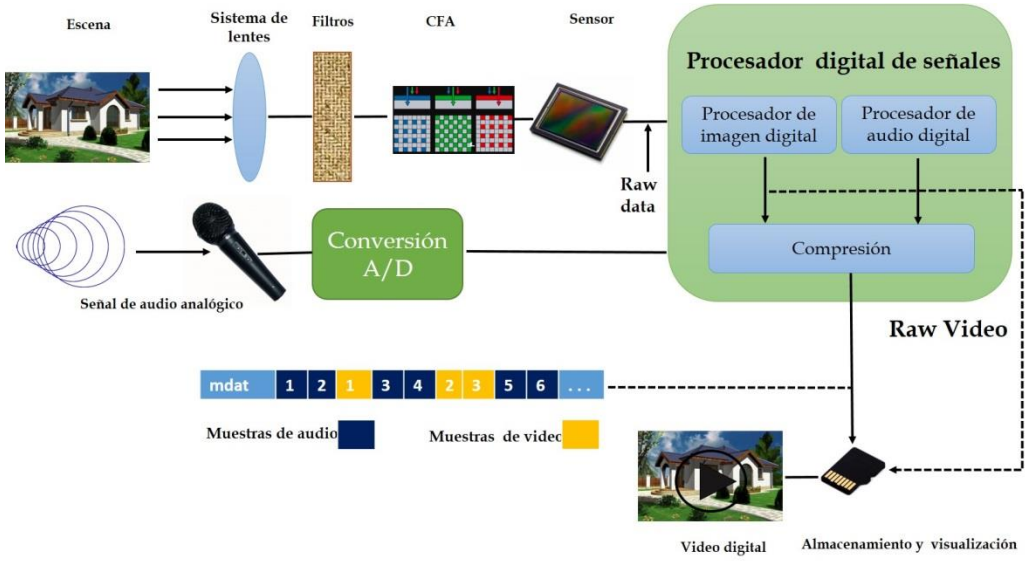


Figura 2.2. Proceso de adquisición de vídeos en cámaras digitales

- **Procesamiento de Secuencia de Imágenes**: Este proceso se inicia cuando el sistema de lentes captura la luz de la escena controlando la exposición, el foco y la estabilización de la imagen. La luz pasa varios filtros para mejorar la calidad visual de la imagen. Como mínimo se aplica un filtro infrarrojo que absorbe o refleja la luz para que pase únicamente la parte visible del espectro a la siguiente fase y evitar pérdida de nitidez en la imagen, y un filtro *anti-aliasing* que limpia la señal generando contornos más suaves en las imágenes. Luego, la luz pasa al sensor de la imagen a través de la matriz de filtro de color. Esta señal se convierte en una señal digital y se transmite al procesador de imagen que forma parte del DSP. La señal digital generada es captada por el procesador de imagen que la somete a diferentes procesos con el fin de estabilizar la señal y corregir alteraciones, por ejemplo, eliminar el ruido y otras anomalías introducidas [Cor12][Nak05]. A continuación, la señal estabilizada pasa al proceso de compresión mediante un códec en concreto que luego se encapsula en un contenedor que soporte estos datos. En las cámaras de dispositivos móviles normalmente se utiliza H.264 o MPEG-4 parte 10 que es la norma que define el códec de vídeo de alta compresión, con un contenedor multimedia MP4. Este contenedor tiene la capacidad de

almacenar o encapsular vídeo y audio.

- **Procesamiento del audio**: Se inicia cuando la señal sonora transmitida por el aire es capturada a través del micrófono. El micrófono transforma las ondas sonoras en una señal eléctrica para aumentar su intensidad y transmitirla a un conversor analógico digital o *Analog Digital Conversion* (ADC) para convertirla en una señal digital. La señal de audio digital la captura el procesador de audio digital para mejorar la calidad del audio antes de la compresión. Después la señal es comprimida con algún algoritmo de codificación para posteriormente ser encapsulado en un contenedor y almacenado en un dispositivo. En su gran mayoría las cámaras de los dispositivos móviles utiliza el algoritmo de compresión con pérdida o *Advanced Audio Coding* (AAC).

Finalmente el fichero con extensión MP4 será guardado en un dispositivo de almacenamiento.

2.2. Compresión del Vídeo

Para reducir la el tamaño de un vídeo se puede utilizar una de las siguientes estrategias: Reducir las dimensiones de los fotogramas, reducir la velocidad de los fotogramas capturados y la compresión o codificación de datos (más usada).

El proceso de compresión o codificación de un vídeo significa convertirlo a un formato adecuado para su transmisión o almacenamiento reduciendo típicamente el número de bits. Un vídeo sin comprimir con una definición estándar requiere aproximadamente 216 Mbits por segundo [Iai10].

El proceso de compresión tiene: un sistema de codificación o compresión convierte los datos recibidos en un formato comprimido usando un número reducido de bits antes de la transmisión o almacenamiento, y un sistema de decodificación que convierte los datos comprimidos a una representación de los

datos de vídeo original. La compresión de datos se consigue eliminando la redundancia (los componentes que no son necesarios para una reproducción fiel).

Los tipos de algoritmos que se utilizan para realizar la compresión de un vídeo son: algoritmo de compresión con pérdida y algoritmo de compresión sin pérdida. El algoritmo de compresión con pérdida se basa en el principio de la eliminación redundancia subjetiva, es decir, los elementos de la secuencia de fotogramas que se pueden quitar sin que afecte de manera significativa la percepción del espectador de la calidad visual. Generalmente se utiliza el algoritmo de compresión con pérdida para comprimir la imagen e información del vídeo. Dependiendo del tipo de algoritmo de compresión utilizado, la calidad de la imagen tendrá perdidas en mayor o menor proporción. La compresión puede estar implementada a nivel de *software* o de *hardware*. Asimismo, existen códecs optimizados orientados a la calidad de los fotogramas y otros a la velocidad de codificación.

Por tanto, para comprimir un vídeo, primero se obtiene la señal digital estabilizada, luego se codifica la imagen a sus componentes originales RGB, *Intensity-Hue-value* (YUV) o cualquier método de almacenamiento de vídeo digital, y por último se aplican los algoritmos que realizan la compresión. Algunas de las técnicas de compresión existentes son las siguientes:

- **Tabla de consulta de color**: También denominada *Colour Lookup Table* (CLUT), es una tabla que almacena la información de color de los píxeles o regiones.
- **Truncamiento**: Reduce el número de bits por cada componente, píxel o resolución, siendo esta técnica la más simple de todas por su mínima complejidad al momento de ser procesado.
- **Interpolación de regiones**: Permite que las regiones tengan cambios logrando almacenar la región por primera vez y, gracias a las técnicas de

interpolación, reconstruir la siguiente región.

- **Transformación de regiones**: Se cambia la información de una región por otra, devolviendo un resultado visual similar. Uno de los algoritmos usados para la transformación es la Transformada Coseno Discreta o *Discrete Cosine Transform* (DCT).

- **Compensación de movimiento**: Usa varias de las técnicas descritas anteriormente, centrándose en encontrar las partes que sufren cambios menores, dividir la imagen en bloques y realizar los cambios necesarios. Cuando los cambios de las regiones son mínimos, procura no hacer cambios o predecir valores. El estándar MPEG aplica esta técnica.

Los Tipos de codificación existentes se presentan en la Tabla 2.1.

Tipo de Codificación	Descripción
Intra o Espacial	Se encarga de comprimir cada imagen de forma independiente dejando de lado los datos de tiempo en el proceso de compresión. Se denomina codificación intra, interna, o espacial. Dentro de esta categoría se encuentra el estándar de compresión *Joint Photographic Experts Group* (JPEG). El vídeo se puede codificar mediante una sucesión de fotogramas codificados inicialmente con JPEG. Esta codificación recibe el nombre de JPEG en movimiento.
Inter o Temporal	aprovecha la similitud de imágenes sucesivas. El codificador inter envía la diferencia entre la imagen previa y la actual en forma de codificación diferencial, para eliminar la redundancia temporal teniendo en cuenta la información de las imágenes previamente enviadas. Luego, envía únicamente las zonas de la imagen que han sufrido cambios de un fotograma a otro. La primera imagen almacenada con anterioridad es denominada fotograma de referencia o *key frame*, y es un elemento necesario para el codificador. Esto se debe a que el fotograma será comparado con los fotogramas siguientes, también es necesario una imagen previamente almacenada para que el codificador genere las imágenes siguientes.

Tabla 2.1: Tipos de codificación

2.2.1. Compresión MPEG

Los dispositivos móviles en su mayoría utilizan el estándar de compresión MPEG. MPEG contiene varios estándares y éstos a su vez se clasifican en partes, que se actualizan o amplían periódicamente para dar soporte a nuevos requerimientos. Este algoritmo de compresión de vídeo utiliza dos técnicas:

- **Codificación DCT**: Para la reducción de la redundancia espacial.

- **Compensación del movimiento basada en bloques**: Para la reducción de la redundancia temporal. Se aplica en ambas direcciones: hacia adelante o causal y hacia atrás o no causal. La señal restante es codificada utilizando las técnicas basadas en transformaciones. Los predictores de movimiento, denominados vectores de movimiento, son transmitidos junto con la información espacial.

Una secuencia de vídeo MPEG es básicamente la salida del material en bruto (flujo de bits) de un codificador y solo contiene lo necesario para que un decodificador restablezca la imagen original. Según [Woo05] la secuencia de vídeo MPEG tiene una estructura en capas bien definidas. Cada capa, contiene la muestra individual, un encabezado que posee fragmentos de metadatos y una alineación del patrón de bits para que se diferencien dentro del flujo de bits.

Un GOP es la unidad fundamental de codificación temporal y una de sus características es especificar el orden de las imágenes. Está representado por secuencias de 10 a 30 fotogramas. Puede contener distintos tipos de imágenes [C.B05]:

- **Imágenes de codificación intra (*I-Frames*)**: Son imágenes de referencia que representan una imagen fija independientes de los otros tipos de imágenes.

- **Imágenes de codificación mediante predicción (*P-Frames*)**: Contienen información de la compensación de movimiento de la imagen precedente, ya sea del tipo *P-Frame* o *I-Frame*.

- **Imágenes de codificación mediante predicción bidireccional (*B-Frames*)**: Contienen información diferente de las imágenes precedente y siguiente.

Un GOP siempre empieza con una imagen tipo *I-Frame*, seguida de cualquier número de *I-Frames* y *P-Frames*, considerados como marcos de anclaje. Entre cada par de fotogramas consecutivos de anclaje pueden aparecer varios *B-Frames* como se observa en la Figura 2.3.

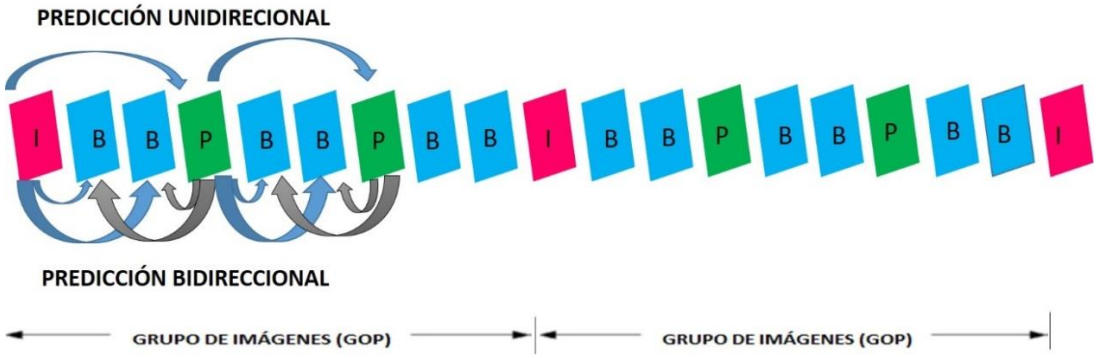


Figura 2.3: Estructura de un grupo de imágenes

Un *I-Frame* no se refiere a ningún otro fotograma de vídeo y, por lo tanto, puede ser decodificado de forma independiente, proporcionando un punto de entrada para un rápido acceso aleatorio al vídeo comprimido. Asimismo, la codificación de un fotograma *P-Frame* se basa en un marco de anclaje anterior, mientras que la codificación de un fotograma *B-Frame* se puede basar en dos marcos de anclaje: uno anterior y uno posterior. Si la información de la imagen I-*frame* se encuentra en RGB debe convertirse a YUV, antes de comenzar el proceso de compresión. Esto es heredado de JPEG y es vital para la compresión ya que el ojo humano es más sensible al componente Y que a las componentes CbCr. Esto permite al algoritmo disminuir la cantidad de datos para la representación del color.

2.2.2. Compresión H.264/MPEG-4 AVC

El estándar H.264/MPEG-4 *Advanced Video Coding* (AVC) es una evolución de la codificación establecida en los estándares MPEG e *International Telecommunication Union* (ITU). Algunas mejoras con respecto a los estándares anteriores son:

1. Uso de estimación de movimiento o *Motion Estimation* para mejorar la predicción entre fotogramas *(inter-picture prediction)* y eliminar redundancias temporales.
2. Uso de la correlación espacial de los datos para realizar la predicción intra fotogramas *(intra-picture prediction)*.
3. La construcción de residuos como la diferencia entre las imágenes predecidas y las imágenes originales. El uso de una transformada espacial discreta y la aplicación de un filtro para eliminar redundancias espaciales y por último la codificación de la entropía de los coeficientes residuales de la transformada y de los datos de apoyo, como los vectores de movimiento que se calculan al usar la estimación de movimiento.

El estándar H.264/AVC se divide en dos partes: Una capa de codificación de vídeo o *Video Coding Layer* (VCL), que representa la codificación y el contenido de vídeo, y la capa de abstracción de red o *Network Abstraction Layer* (NAL), que adapta la información que proviene de VCL, que posteriormente le proporciona un formato. En este trabajo se hace énfasis en la capa de abstracción de red que es necesaria para realizar la extracción de fotogramas presentado en capítulos posteriores.

2.3. Importancia del Análisis Forense

La necesidad de realizar un análisis forense en dispositivos móviles surge a partir de las bondades y características tecnológicas que pueden ofrecer este tipo de dispositivos. Por ejemplo, la facilidad de implementar aplicaciones para

estos dispositivos en poco tiempo [AZ06], la capacidad de almacenar, editar, eliminar e imprimir documentos electrónicos, el uso de mensajes de texto, mensajes multimedia y conversaciones a través de aplicaciones de redes sociales (lo más requerido en la actualidad), el uso de plataformas en línea para realizar operaciones bancarias, compras a través de la web y todas aquellas operaciones realizadas con datos sensibles. Por tanto, hoy en día, los dispositivos móviles son elementos que proveen información útil que deberá tratarse con cautela y precisión.

2.4. Uso de Vídeos en el Análisis Forense

Partiendo del hecho que en el mundo hay más teléfonos móviles que personas y que gran parte son smartphones que graban vídeos, es ampliamente conocido que la mayoría de dispositivos móviles actualmente tienen sistema operativo Android, y particularmente en EEUU son IPhones. Sin miedo a equivocación, prácticamente el 100% de los dispositivos móviles lleva una versión de Android igual o superior a 3.0. Esto lleva a pensar que el estándar de empaquetado y compresión de video es el H264 MP4 [Int16].

En cierto sentido, es una suerte que MPEG4 siga utilizándose, ya que tiene rasgos comunes con MPEG2 [Int96]. Investigadores y profesionales han realizado estudios e implementaciones de dicho estándar y desarrollado técnicas forenses de caracterización de dichos vídeos digitales, con el objetivo de facilitar su trazabilidad, garantizar su integridad, entre otras cosas. Porque al fin y al cabo, los videos no son más que otra pieza de información. Cualquier tipo de análisis forense al que pueda someterse, por ejemplo, un documento PDF, debe tener su equivalente para videos.

La mayoría de herramientas de edición de vídeo existentes no funcionan directamente en el dominio comprimido. Por lo tanto, el proceso de edición de una secuencia de vídeo se compone de tres pasos principales: Decodificación de

la secuencia de entrada, edición del vídeo real y re-codificación del vídeo editado. Conocer y entender este proceso es muy útil para el análisis forense ya que permite identificar posibles manipulaciones. Algunos investigadores ya están utilizando este proceso para tal fin.

2.5. Técnicas de Análisis Forense

En el caso del desarrollo de técnicas de análisis forense en vídeos digitales, existe poca literatura al respecto. Según [BFM+12a], las técnicas forenses en vídeos se agrupan en herramientas forenses según el objetivo a cumplir: herramientas forenses para el análisis de adquisición, compresión y autenticación de un vídeo. Algunos trabajos se basan directamente en la secuencia de codificación y otras en la extracción de fotogramas aplicando a algún método de clasificación para imágenes fijas [RP07] [MTT12].

Hay que tener en cuenta que la integridad y autenticidad de un vídeo no puede garantizarse al 100% ni mucho menos. Sobre todo porque la compresión de video genera muchísimo ruido, por no hablar de la propia compresión intra-fotograma.

Para un vídeo sospechoso, el vector local de movimiento local es usado para segmentar regiones de fondo, en cada fotograma. En función de la segmentación de los fondos, y la cantidad de movimiento del primer plano, la secuencia de predicción modificada residual es calculada, y esta tiene huellas consistentes de una posible doble compresión. Después del post-procesado, los resultados de detección y estimación GOP, son obtenidos al aplicar el análisis temporal periódico a la secuencia generada.

3. ESTADO DEL ARTE

A pesar de ser más complicado que para las imágenes, crear un vídeo falsificado ahora es más sencillo que antes, debido a la disponibilidad de herramientas de edición de vídeos. Al mismo tiempo, los vídeos son ampliamente utilizados para la vigilancia, y por lo general se consideran una prueba mucho más fuerte que una sola toma de vídeo. Hay diferentes maneras de manipulación de un vídeo, y algunos de ellos no son complicados en absoluto: uno puede estar interesado en la sustitución o eliminación de algunos fotogramas (por ejemplo, de una grabación de vídeo-vigilancia), duplicar un conjunto de fotogramas, introducir, o eliminar algunos de los objetos de la escena.

Es posible identificar técnicas de anti- forense de vídeo como la falsificación de un vídeo por medio de la redundancia espacial y temporal. La redundancia espacial o intra se realiza tratando cada fotograma de forma independiente. La redundancia temporal considera la relación que existe entre fotogramas adyacentes. Aunque sería posible analizar la integridad de un vídeo aplicando las herramientas forenses de imágenes, este enfoque se considera poco práctico por las siguientes razones:

- **Complejidad**: Las herramientas para la detección de falsificaciones en las imágenes por lo general tienen elevada complejidad de procesamiento.
- **Fiabilidad**: La copia o eliminación de fotogramas no sería detectado por ninguna herramienta forense de imágenes.
- **Conveniencia**: la creación de vídeos que son manipulados que son constantes en el tiempo son muy difíciles de detectar por lo que este tipo de relaciones entre los fotogramas son un activo valioso para la identificación de la falsificación.

La doble compresión de tipo 1 consiste en que el vídeo manipulado, sigue la misma estructura GOP que el original. Por lo tanto, nos encontramos con fotogramas-I que han sufrido una doble compresión, muy parecida a la doble compresión JPEG. La doble compresión de tipo 2, implica que el video manipulado no sigue la misma estructura GOP que el original. Por lo tanto algunos fotogramas-I pasan a ser fotogramas-P. Por lo tanto, los métodos para tipo 1 ofrecen un rendimiento pobre al ser aplicados a material re-comprimido de tipo 2. Hay varios métodos para detectar la doble compresión tipo 2, entre los que pueden mencionarse los siguientes:

En [SLL12] se desarrolló un método automatizado, comprobándolo con técnicas anti-forenses. En [WF06] se usa la periodicidad de la secuencia residual media de predicción. En [VPFB+12] se propuso una técnica basada en la predicción mediante variación de macrobloques en los fotogramas-P recodificados. Es parametrizable para varios grados de compresión y es capaz de detectar el tamaño de GOP inicial.

En [LWH08] se tuvo en cuenta los errores de bloque de los vídeos re-comprimidos, al borrar diferentes fotogramas de una secuencia dada. Aun así, este método no es automatizable, necesita de un ser humano para dar el visto bueno final.

En [SSCL15] se propusieron dos métodos forenses posibles, que pudieron derrotar los métodos anti-forense propuestos en videos H.264. EL primer método consiste en emplear el filtro de desbloqueo de H.264/AVC y el segundo método está relacionado con la selección del parámetro de cuantización.

La detección de re-compresión de tipo II es complicada, ya que el tamaño GOP del video original es desconocido. Los estudios existentes acerca de detección de re-compresión de tipo II se centran tanto en el dominio comprimido como en el descomprimido. Pero muy pocos se fijan en el contenido comprimido y el contenido visible al usuario.

En [BFM+12b] se propone un método para identificar las regiones manipuladas en MPEG- 2 con sólo I-Frames. Posteriormente, los mismos autores proponen tener en cuenta la información sobre el error de movimiento al utilizar P-Frames, a fin de detectar eliminación o adición de marcos.

En [SWJ12], estiman el parámetro de cuantificación y los vectores de movimiento de marcos decodificados.

En [JWS+13] se propone un nuevo enfoque para la identificación de una secuencia de vídeo que se ha codificado doblemente. Este método funciona al recomprimir el vídeo bajo análisis con los tres posibles códecs y al calcular una medida de similitud entre las dos secuencias.

En [LBR+13] se propone un método para detectar si un vídeo se ha codificado dos veces, si este es el caso, estima el tamaño del grupo de imágenes (GOP) empleado durante la primera codificación. En la propuesta se utiliza una huella robusta y muy distintiva basada en la variación de los tipos de predicción de macrobloques en los P-frames recodificados. Una ventaja de esta variación de predicción de huella (VPF) es su presencia en el vídeo codificado dos veces sin la necesidad de re-compresión. Por otra parte, teniendo en cuenta que la VPF se hace evidente sólo en P-Frames que se intracodifican en la primera codificación no consideran el uso de B-Frames. Para tal fin, se limita la compresión que se realiza de acuerdo al perfil de base para H.264 y al perfil equivalente para MPEG-2 y MPEG-4. Estos perfiles de soporte sólo están en I-Frames y P-Frames, junto con los tres principales tipos de macrobloques: intra-codificado (I-MB), inter-codificados (P-MB) y omitidos (S-MB). Un I-Frame codificado sólo puede contener macrobloques I-MB, mientras que P-Frames codificados pueden contener cualquiera de los macrobloques mencionados, es decir, I-MB, P-MB o S-MB. En general, la matriz de cuantificación o el factor de calidad para codificar un I-Frame difieren de la considerada para un cuadro P-Frame debido a que los I-Frames se utilizan de forma directa o indirectamente

como referencia para codificar varias tramas futuras. Se concluye que si se puede detectar esas variaciones en el número de tipos de predicción I-MB y S-MB, entonces se puede de detectar si una doble codificación de la misma secuencia ha sido llevada a cabo y, si este es el caso, se puede estimar el tamaño de la primera GOP de esas variaciones. La VPF se puede utilizar para detectar doble codificación y para estimar el tamaño de GOP de la primera compresión

En [VPFB+12] se aborda la localización de falsificación en vídeos comprimidos en MPEG-2. El método propuesto se basa en el análisis de Doble de Cuantificación (DQ) que se traza en los I-Frames. Estos marcos se encuentran en el vídeo bajo análisis pudiendo estimar con ellos el tamaño del grupo de imágenes (GOP) que era utilizado en la primera compresión. Posteriormente, el análisis DQ se ideó para el esquema de codificación MPEG-2 y se aplica a los marcos que fueron intra-codificados tanto en la primera y segunda compresión. De tal manera, que las regiones que fueron manipuladas entre las dos decodificaciones se detecta. En comparación con los métodos existentes basados en el doble análisis de cuantificación, el esquema propuesto hace posible la localización de falsificaciones en una amplia gama de configuraciones.

Este método permite determinar qué partes de un marco han sido alteradas. El método funciona básicamente mediante la búsqueda de rastros de doble cuantificación a nivel espacial, lo que permite la construcción de un mapa de grano fino de probabilidad de manipulación para cada marco analizado. Se centran en el escenario de falsificación con marcos intra-codificados, y suponen que, a partir de una secuencia de vídeo MPEG-2, el atacante decodifica el vídeo, altera la contenido de un grupo de tramas, y finalmente se codifica nuevamente la secuencia resultante usando MPEG-2 con un tamaño diferente.

En [Int96] se combinan técnicas de estegoanálisis como el análisis de características de densidad conjunta de los elementos circundantes, con la densidad marginal del dominio DCT, para mejorar la detección de doble

compresión JPEG. Asimismo, se estudia la relación entre el factor de compresión, complejidad de la imagen y exactitud de detección. Por otro lado, se pretende aprovechar dichos datos para identificar la fuente de las imágenes producidas por dispositivos móviles (*smartphones*). A primera vista, con el ruido del sensor, se debería poder otorgar cierto poder de clasificación de una imagen, a pesar de ello, las imágenes son objeto a interpolación, recorte, y recompresión. Por lo que se hace necesario buscar otras características, para complementar dicha técnica.

Los autores indican que una imagen al ser una señal multimedia, se puede modelar mediante una distribución Gaussiana Generalizada (GGD), que tiene dos parámetros alfa y beta, que varían adaptativamente. Alfa mide la anchura del pico de la distribución, mientras que beta modela la forma de la distribución. En compresión JPEG, un bloque DCT tiene 64 (8x8) coeficientes de frecuencia. Por tanto, se puede decir que la densidad marginal de dichos coeficientes sigue la distribución GGD, y que manipulaciones como doble compresión cambian dicha densidad marginal al aplicar dos veces tablas de cuantización. Adicionalmente, proponen usar un modelo distribución Gaussiana Generalizada multi-parámetro para evaluar las densidades conjuntas

Por regla general, la manipulación a imágenes JPEG, modificará los coeficientes DCT y su densidad marginal, en sus respectivas coordenadas de frecuencia. (en la tabla de cuantificación de JPEG, los valores más altos están agrupados en el fondo derecho de las coordenadas de alta frecuencia, a fin de meter muchos ceros en los coeficientes DCT de alta frecuencia. En otras palabras, los coeficientes DCT no nulos están agrupados en coordenadas de baja frecuencia. Por ejemplo, las modificaciones ocurren en bandas de baja frecuencias. El algoritmo propuesto primero comprueba que el modelo se verifica dentro de un mismo bloque y que el modelo se cumple para densidad conjunta inter-bloque. En los resultados, la tasa de detección de este método es significativamente buena. Sin embargo, en el caso de videos, objeto de

investigación de este Trabajo Fin de Grado dichos métodos de detección no son muy aplicables. Esto se debe a la limitación propia embebida en el dispositivo del bitrate de un video, que puede forzar matrices de cuantificación, que producen auténticas carnicerías, en ciertos fotogramas de vídeo.

Vídeos grabados por cámaras fijas son ampliamente utilizados en vídeovigilancia y videoconferencias. Este tipo de vídeos, a menudo, tiene un fondo estático, o que cambia lentamente, de manera gradual. En estos vídeos de fondo estático, hay una redundancia temporal muy pronunciada. Para este tipo de video, la operación de manipulación es más fácil, ya que entre un fotograma y otro hay pocos cambios. Por lo tanto, En [HJSW16] se propone un método de detección basado en la detección de movimiento local en cada unidad GOP. Dicha detección se hace para los macrobloques. Cabe mencionar que la doble compresión es lo más parecido, a lo que en herramientas comerciales se llama 'compresión de dos pasos', en el que se crea una estructura inicial de fotogramas, y en el segundo paso se intenta rebajar aún más el bit-rate, eliminando algunos fotogramas I, para codificarlos como Fotogramas-P. El método aprovecha la existencia de macrobloques, y la distribución de los vectores de movimiento, para estimar el tamaño de GOP original. SI el tamaño de GOP original estimado difiere del del video existente, entonces dicho vídeo ha sufrido una doble compresión.

En [HHLH08] **se** propone un nuevo método para localizar las regiones manipuladas de un vídeo, utilizando correlación de residuo de ruido. Se modela la distribución de correlación de ruido como modelo de mezclas gaussianas (GMM) y utiliza un clasificador bayesiano para encontrar el valor del umbral óptimo basado en los parámetros estimados. El método propuesto consiste en dos pasos para para deducir los parámetros de dicho modelo. En el primero, el residuo de ruido de cada fotograma es extraído al restar el fotograma filtrado (con un filtro anti-ruido) del fotograma original. Se propone usar un filtro anti-ruido wavelet. En el paso 2, cada fotograma es particionado

en NxN bloques. La correlación de residuo de ruido entre bloques de mismo índice espacial de dos fotogramas consecutivos. Finalmente, localiza los bloques manipulados analizando las propiedades estadísticas de correlaciones de ruido a nivel de bloque. El método tiene limitaciones con los vídeos muy 'movidos', por lo que pueden aparecer regiones con ruido inconsistente y con vídeos de bajo bitrate.

Como las cámaras de vídeo suelen dejar una huella característica en los vídeos grabados. Aunque este tipo de defectos suelen ser explotados sólo para la identificación de dispositivos. Las principales contribuciones en este campo son de Mondaini [CGS98], Hsu [MPFL], y Kobayashi [WHL12].

Mondaini [CGS98] propuso una aplicación directa de la técnica de huellas del patrón de ruido PRNU. El patrón característico de una vídeocámara se estima en los primeros fotogramas de vídeo, y se utiliza para detectar varios tipos de ataques. Específicamente, los autores evalúan tres coeficientes de correlación. El primero entre el ruido de cada fotograma y el ruido de referencia, el segundo entre el ruido de dos fotogramas consecutivos y el tercero entre los fotogramas sin extracción de ruido. Cada uno de estos coeficientes de correlación es el umbral para obtener un evento binario, y diferentes combinaciones de eventos permiten detectar diferentes tipos de manipulaciones, entre los que destacan: inserción de un fotograma, insertar objetos en un fotograma mediante el ataque de cortar/ pegar y repetir fotogramas. Los experimentos se realizan en vídeos sin comprimir y comprimidos con la técnica *MPEG* los resultados muestran que el método es fiable en los vídeos sin comprimir (sólo se informa de algunos estudios de casos, no los de los valores en promedio), mientras que la codificación MPEG afecta al rendimiento de forma significativa.

Hsu [MPFL] adopta una técnica basada en la correlación temporal de los residuos de ruido, donde el "residuo de ruido" de un fotograma se define como

lo que queda después de restar del fotograma su versión sin ruido (utiliza la técnica de filtración propuesta en [WBSH09]). Cada fotograma se divide en bloques, y se evalúa la correlación entre el residuo de ruido de bloques vecinos en el espacio temporal (es decir, bloques en la misma posición que pertenecen a dos fotogramas adyacentes). Cuando se falsifica una región, el valor de correlación entre los residuos de ruido temporal cambiará de forma radical: se verá reducida si los píxeles de los bloques se pegan desde otro fotograma/región, mientras que se elevará a 1 si se produce una repetición del fotograma.

Los autores proponen un enfoque de detección de dos etapas para reducir la complejidad del esquema: primero una decisión umbral aproximado/en bruto se aplica a las correlaciones y, si el fotograma contiene un número significativo de bloques sospechosos, se realiza un análisis estadístico más profundo para modelar el comportamiento de correlación del residuo del ruido a través de una mezcla Gaussiana y la estimación de sus parámetros. Los resultados son modestos, cuando se utiliza la técnica de copiar y pegar para atacar vídeos, en promedio solo se detectan el 55% de los bloques falsificados (la tasa de fallos positivos es del 3,3%). Cuando se trabaja en fotogramas "pintados sintéticamente" la detección se eleva al 74% pero la tasa de fallos positivos aumenta también, siendo ésta de un 7% en promedio. Además, cuando el vídeo se codifica con pérdidas, estos resultados caen rápidamente con la fuerza de cuantificación. Sin embargo, a pesar de que los autores no proporcionan experimentos en este sentido, este método debe ser eficaz para la detección de la repetición de fotogramas, que es un ataque importante en el escenario de vídeo vigilancia. Vale la pena señalar que, a pesar de las características de la cámara que se tratan, este trabajo no se centra en la toma de huellas digitales del dispositivo.

Otro enfoque basado en la cámara es la de Kobayashi [WHL12]: proponen para detectar regiones sospechosas en el vídeo grabado de una escena estática

mediante el uso de características de ruido del dispositivo de adquisición. En concreto el ruido de los fotones del disparo se explota, porque depende principalmente de la radiación a través de una función llamada función de nivel de ruido (*Noise Level Function, NLF*). El método calcula la probabilidad de falsificación para cada píxel mediante la comprobación de la consistencia de la función de nivel de ruido de regiones falsificadas y regiones no falsificadas. Dado que no se sabe a priori que píxeles pertenecen a una región, la EM se emplea [Li10] algoritmo para estimar simultáneamente el NLF para cada fuente de vídeo y la probabilidad de falsificación para cada píxel. El núcleo de la técnica reside en estimar correctamente la función de las fluctuaciones temporales de los valores de los píxeles, y esta estimación se discute a fondo desde un punto de vista teórico.

4. THEIA: HERRAMIENTA PARA EL ANÁLISIS FORENSE DE IMÁGENES Y VÍDEOS DIGITALES

En este capítulo se presenta Theia, una herramienta para el análisis forense de imágenes y vídeos digitales que facilita la extracción y el tratamiento de metadatos Exif en imágenes JPEG y átomos en vídeos MP4. A grandes rasgos la herramienta se divide en dos grandes partes: Tratamiento individual y tratamiento masivo de imágenes y vídeos.

4.1. Tratamiento a nivel individual

Permite obtener la información Exif detallada de una imagen individual, situar la imagen en Google Maps y Google Earth (si posee información de geoposicionamiento). La estructura general se puede observar en la Figura 4.1.

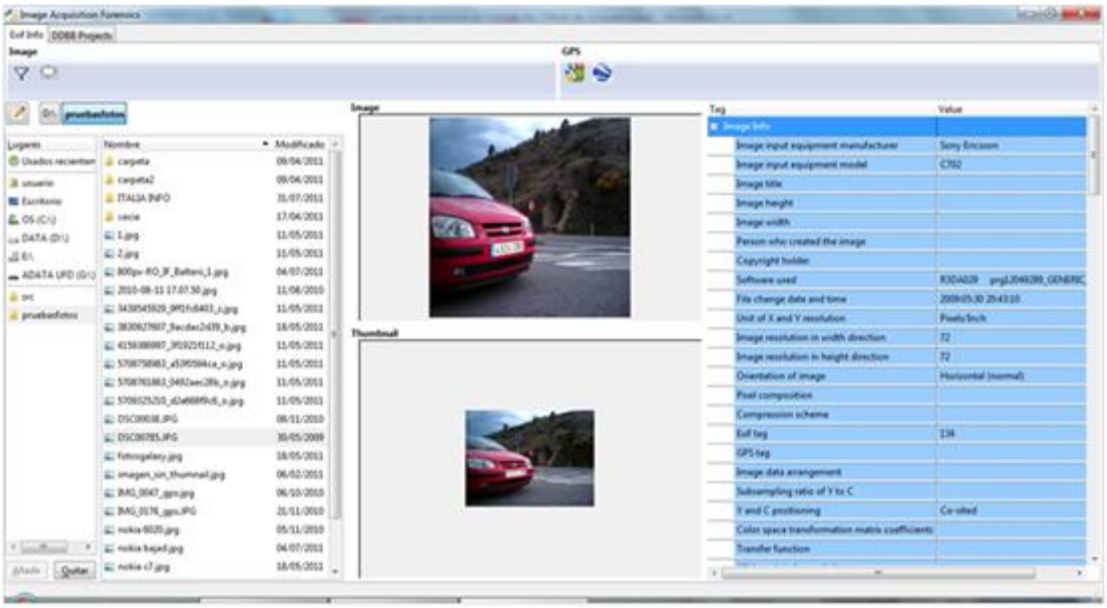


Figura 4.1. Apariencia general de la pestaña *Exif Info*

Como estructura general se puede apreciar a la izquierda de la imagen un navegador de archivos, en el centro la imagen del archivo seleccionado y su correspondiente thumbnail (es el incluido en el propia archivo de la imagen no ninguna generación propia del programa) y a la derecha las etiquetas Exif con

su correspondiente información. De esta estructura cabe destacar en la interfaz gráfica que es totalmente configurable a nivel de tamaños, es decir todos los separadores entre las distintas zonas se pueden mover.

La información Exif se ha organizado en 6 grupos: *Image*, *Exif*, *GPS*, *Interoperability*, *Thumbnail* y *Maker Note*.

- ***Image Info***: En este bloque se almacenan las etiquetas con información relativa a la propia imagen y que no tienen relación directa con el entorno y el momento de la captura. Por ejemplo la marca y modelo de la cámara, el tamaño de la imagen, la unidad utilizada en la resolución X e Y, etc.

- ***Exif Info*:** En este bloque se guardan las etiquetas con información relativa al momento o al entorno de la toma de la imagen. Dentro de este bloque se encuentra por ejemplo la información referente al flash, hora de toma y generación de la imagen, configuración de la lente, etc.

- ***GPS Info***: En este bloque está toda la información relativa al geoposicionamiento. Por ejemplo información de latitud, longitud, altitud, el estado del receptor GPS, etc.

- ***InterOperability Info***: En este bloque se incluyen las etiquetas relativas a la información de las reglas de interoperabilidad, como pueden ser *Exif R98*, *DCF thumbnail file* o *DCF Option file*.

- ***Thumbnail Info***: En este bloque se encuentran todas las etiquetas relativas a la información de *thumbnail*. Por ejemplo su tamaño en vertical y horizontal y el esquema de compresión utilizado.

- ***Maker Note Info*:** Es una etiqueta individual que almacena la información que cada fabricante puede insertar de forma opcional y que no ha sido recogida en ninguna etiqueta Exif.

El formato de esta información es libre y no tiene una estructura prefijada, cada fabricante utiliza la suya propia que incluso puede ser diferente para distintos modelos de la misma marca. Por tanto se muestra como una secuencia de bytes (en hexadecimal). Si se conoce la estructura estos bytes pueden ser decodificados de forma manual.

Asimismo, se pueden extraer los datos de geoposicionamiento que estén incluidos en las imágenes. Si la imagen no tiene la suficiente información para poder ser mostrada en alguna de las opciones al pulsar la opción de geoposicionamiento se mostrará un mensaje indicándolo (*Not enought GPS information*). El geoposicionamiento se puede realizar desde dos opciones: posicionamiento en Google Maps y en Google Earth. En la primera se abrirá el navegador web por defecto del sistema operativo y se mostrará la ubicación inserta en los metadatos de la imagen en un mapa de Google Maps (es necesario conexión a internet). La Figura 4.2 muestra un ejemplo de geoposicionamiento en una fotografía en Google Maps.

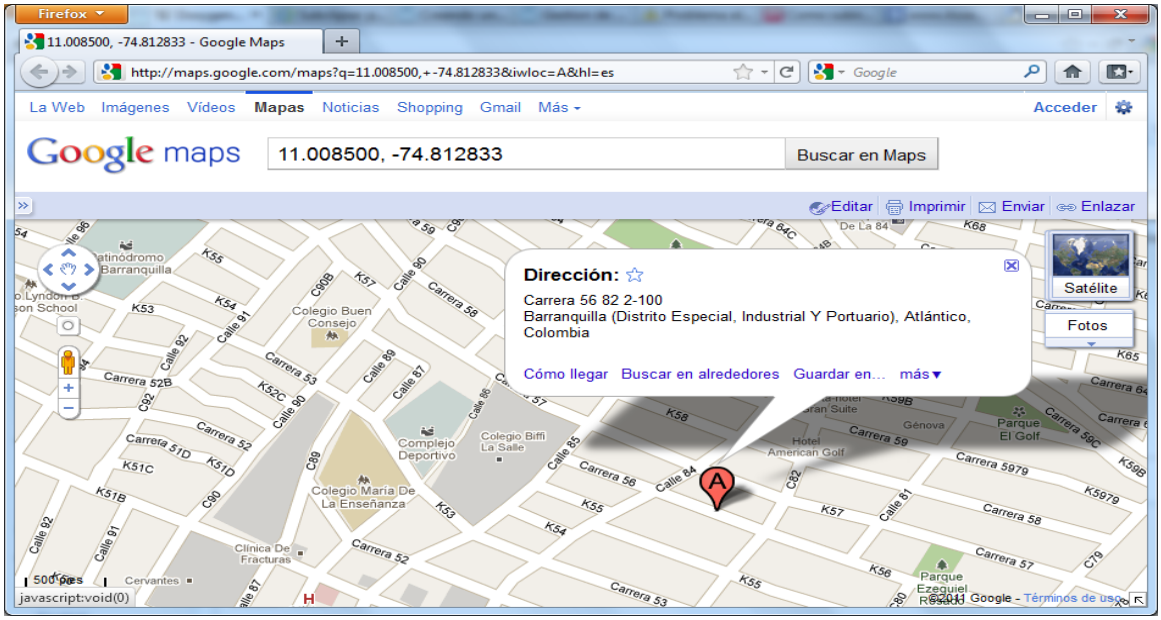


Figura 4.2. Geoposicionamiento en Google Maps

En la segunda opción se abrirá un menú para poder almacenar un archivo de extensión “kml”. Este archivo podrá ser posteriormente abierto si está instalada la aplicación Google Earth, en la cual se mostrará igualmente la posición geográfica almacenada en los metadatos de la imagen (es necesario conexión a internet).

4.2. Tratamiento a nivel de grupo

Permite hacer análisis de imágenes en grupo. Cada grupo es totalmente independiente entre sí. Su apariencia gráfica general puede verse en la Figura 4.3.

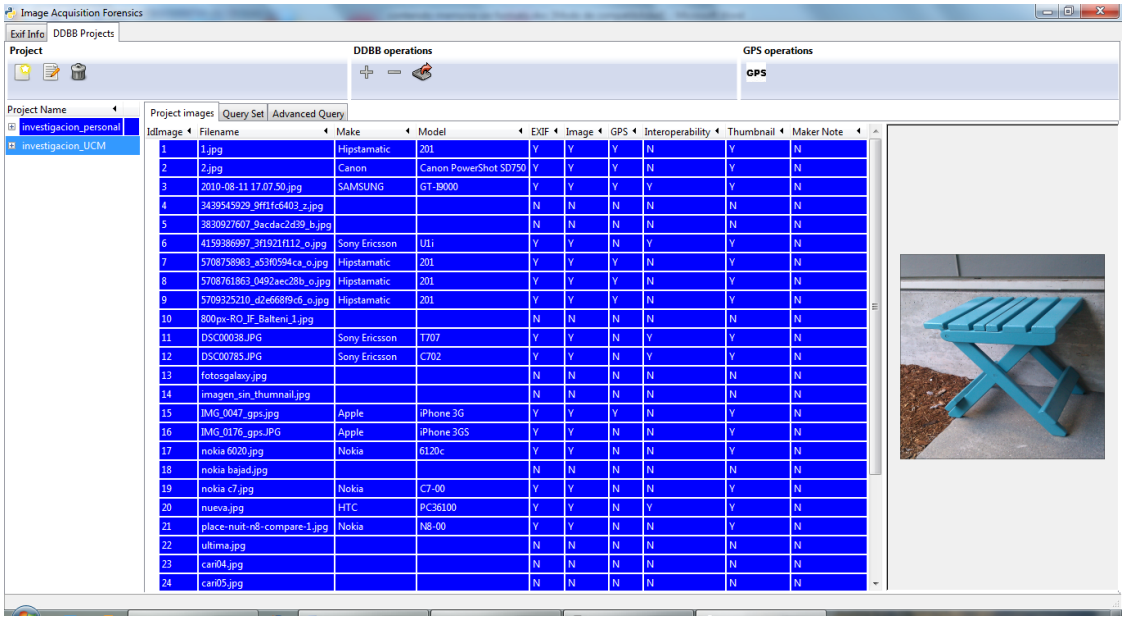


Figura 4.3. Apariencia general de la pestaña *DDBB Projects*

Lo primero a destacar en esta funcionalidad es que las imágenes se tratan en grupos llamados proyectos. Estos grupos pueden ser de una o más imágenes. Cada proyecto es totalmente independiente entre sí. Se busca acercar la realidad del día a día del analista forense a la herramienta, es decir, el analista tendrá diversos casos de análisis disjuntos los cuales podrá tratar en proyectos distintos.

En la parte central de la pestaña *DDBB Projects* y dentro de ésta en la pestaña *Project Images* se muestran una lista de las imágenes del proyecto seleccionado en la lista de proyectos.

Para cada imagen se muestra su identificador interno de la base de datos (para permitir el caso de archivos con el mismo nombre), el nombre del archivo, la marca y el modelo de dispositivo que la creó (si existe). Además se presenta la información de si posee metadatos en los distintos grupos Exif que analiza la herramienta. Asimismo se visualiza el contenido de cada una de las imágenes a la derecha según se van seleccionando. Un ejemplo de captura de esta funcionalidad se muestra en la Figura 4.4.

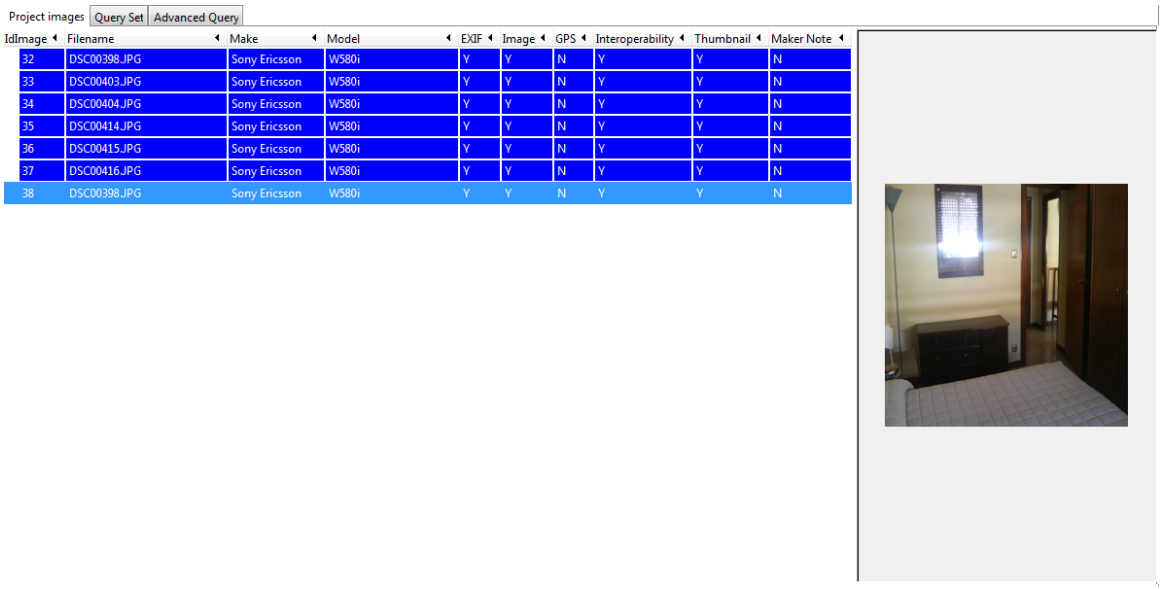


Figura 4.4. Visualización de las imágenes de un proyecto

Los diferentes análisis que se pueden realizar sobre cada proyecto son los siguientes: administración de imágenes (añadir y eliminar imágenes), consultas preestablecidas, consultas avanzadas y geoposicionamiento de las imágenes.

- *Consultas preestablecidas*: Permite crear consultas agregando etiquetas Exif (y otras adicionales que añade la aplicación que ayudan al análisis forense) sobre las imágenes del grupo seleccionado. La consulta agrupa las imágenes por los criterios seleccionados y muestra el número de imágenes que hay en

cada uno de los grupos formados, como puede verse en un ejemplo en la Figura 4.5. En las consultas permiten escoger 5 campos de agregación como máximo (por defecto se realiza sobre *Make* y *Model*, aunque ser pueden elegir cualesquiera). Para escoger los distintos campos hay que pulsar sobre el botón *Query Set*

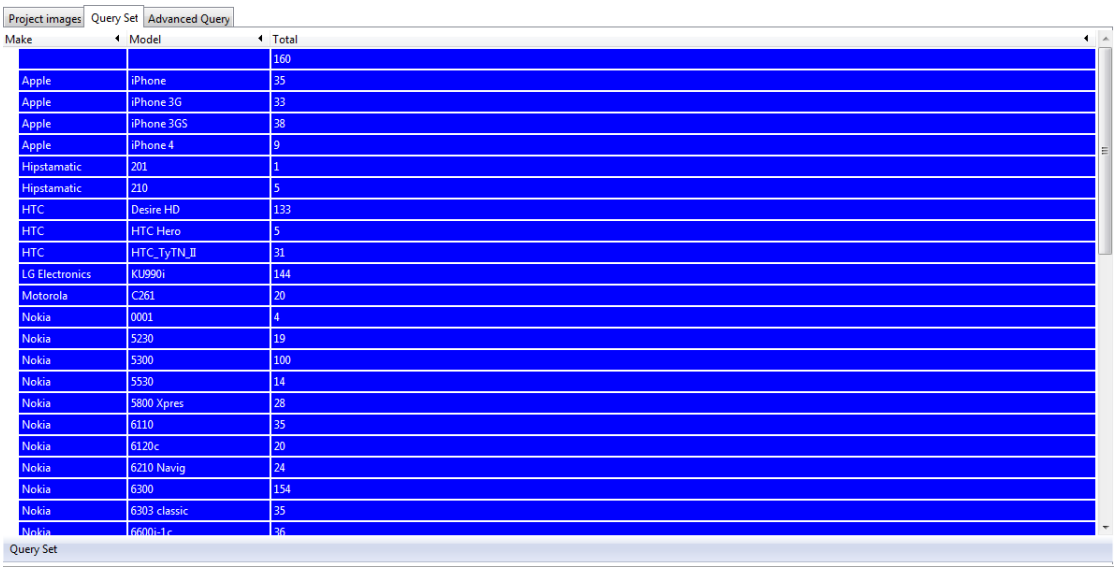


Figura 4.5. Query Set

- *Consultas avanzadas*: Permite la creación de consultas sobre imágenes de un grupo configurando los datos Exif a mostrar y los filtros a aplicar. Es decir, muestra la información de las imágenes de los campos seleccionados que coincidan con uno de los valores de cada uno de los filtros configurados. Asimismo, se permite el almacenamiento permanente de consultas. Una visión general se muestra en la Figura 4.6.

  En *Advanced Query* hay que distinguir dos grandes bloques: la configuración de la consulta y su almacenamiento. Con respecto a la configuración de la consulta avanzada hay que tener en cuenta la configuración de las columnas de los resultados y la configuración de los filtros. En esta consulta se muestran los valores de los campos seleccionados por la configuración de las columnas de los resultados que cumplen las restricciones indicadas en la configuración de los filtros.

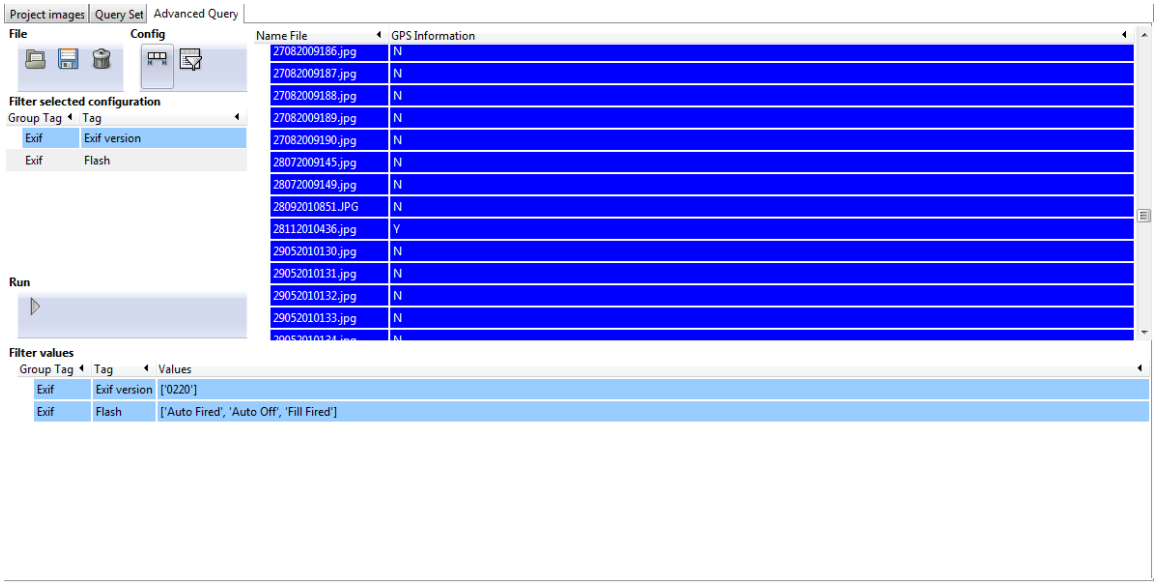
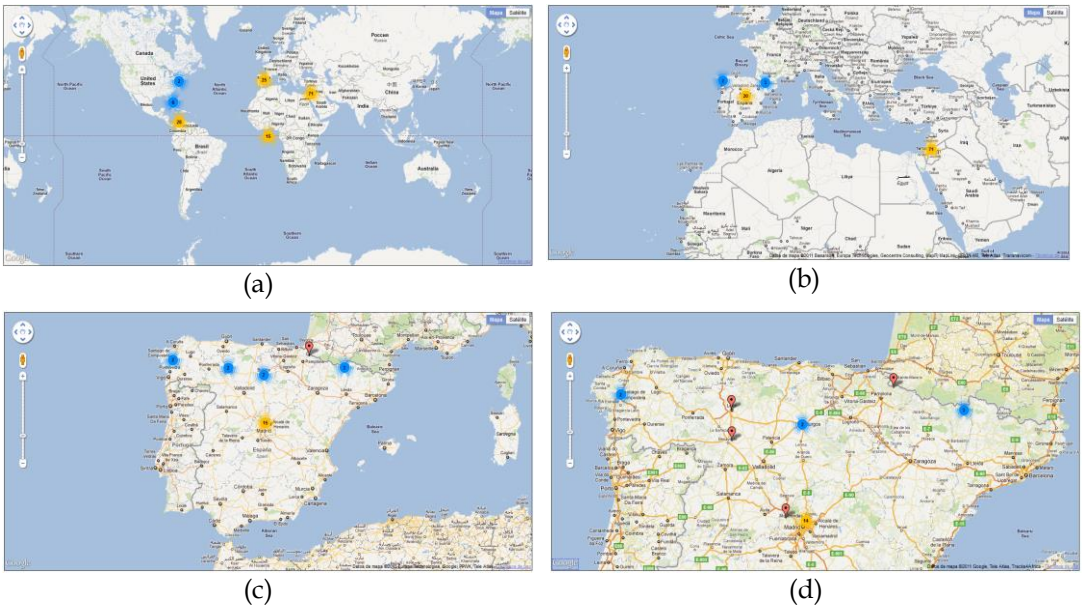


Figura 4.6. *Advanced Query*

- *Geoposicionamiento*: Análogamente al tratamiento de imágenes a nivel individual, existe una funcionalidad que permite el tratamiento de la información de geoposicionamiento para un grupo de imágenes. Esta opción permite la selección de algunas o de todas las imágenes de un grupo con información de geoposicionamiento para la creación de un mapa en Google Maps que sitúe a las mismas. En el mapa se agrupan las imágenes por zona y, a medida que se aumenta el *zoom*, se van detallando las coordenadas. La Figura 4.7 muestra un ejemplo del mapa generado y el proceso de aumento del *zoom* en una zona concreta (desde la Figura 4.7 (a) hasta la Figura 4.7 (d)).



(a) (b) (c) (d)

Figura 4.7. Geoposicionamiento de un grupo de imágenes en Google Maps

5. CONTRIBUCIÓN

El objetivo de este capítulo es presentar la contribución de este trabajo que consiste en un algoritmo de comprobación de integridad de los fotogramas de un vídeo de dispositivo móvil. Por razones de confidencialidad del proyecto se ha omitido información del trabajo desarrollado para no infringir la normativa correspondiente. El algoritmo ha sido implementado teniendo en cuenta los siguientes objetivos:

- Optimización de costes
- Eficiencia
- Multi-procesamiento y/o procesamiento distribuido
- Portabiildad
- Modularidad

5.1. Funcionamiento

El algoritmo tiene como finalidad calcular el porcentaje de confiabilidad de un video.

El algoritmo tiene como objetivo analizar cada fotograma de un vídeo para identificar alteraciones o modificaciones realizadas en el contenido del fotograma. La verificación en cada fotograma se basa en la extracción del ruido del sensor de cada fotograma. El algoritmo se compone, a su vez, de tres procesos: (1) Extracción de fotogramas, (2) extracción del residuo del ruido y (3) análisis mediante umbral y métodos estadísticos. En la Figura 5.1 se presenta el esquema general de funcionamiento del algoritmo.

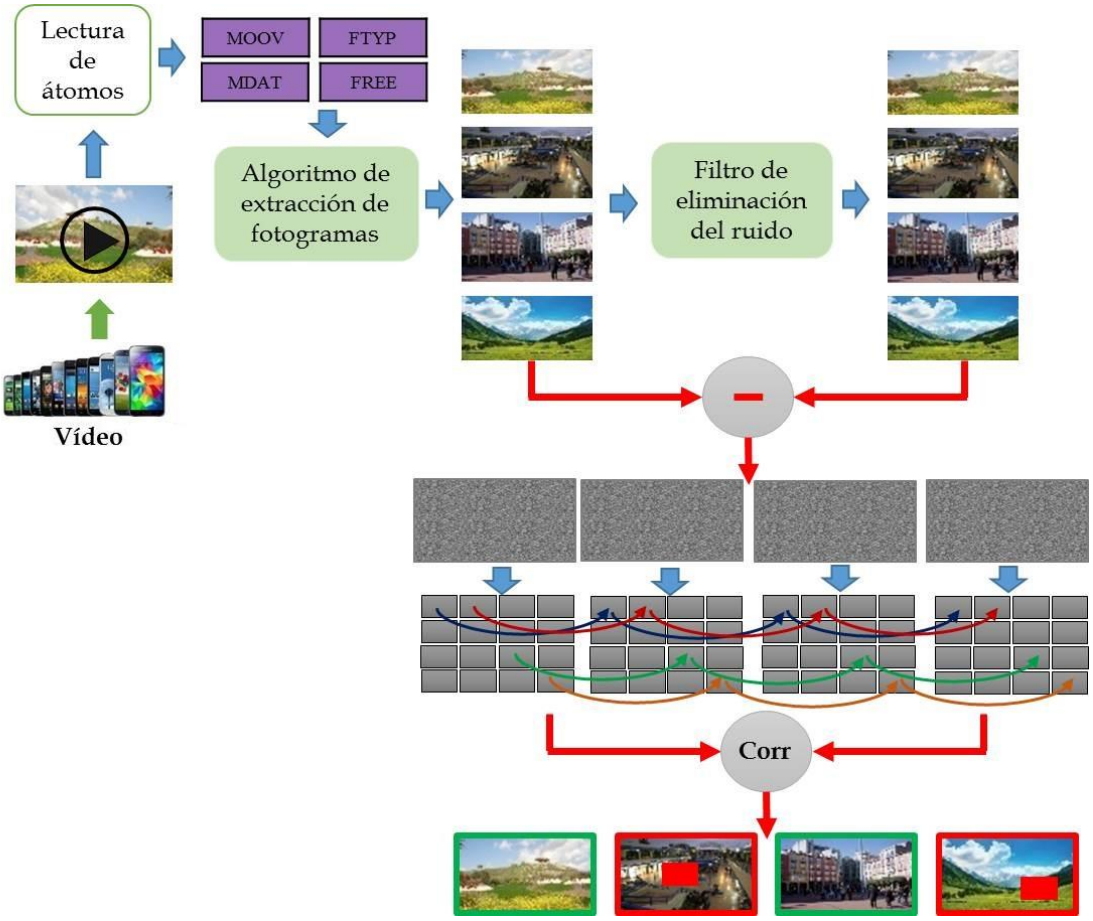


Figura 5.1 Esquema general de funcionamiento del algoritmo

El algoritmo comienza con la lectura del vídeo que se desea analizar. Se verifica que el formato del mismo sea MP4 para extraer los datos necesarios de los átomos *mdat* y *trak* de la pista de vídeo. Los átomos necesarios de la pista de vídeo son: *stsd*, *stsc*, *stco* y *stsz*. Los datos necesarios de cada átomo son los siguientes:

- *stsd*: Contiene los datos de registro de configuración del decodificador que forma parte del átomo avcC.
- *stsc*: Contiene el número de muestras en un fragmento.
- *stco*: Contiene la ubicación del fragmento. Este desplazamiento se conoce desde el principio del archivo.
- *stsz*: Contiene el tamaño de cada muestra de un fragmento.

Se obtiene el número de entradas del átomo *stsc* y el número de muestras en cada fragmento del diccionario *entries*.

De cada fragmento se obtienen tres datos importantes: el número de fragmento, el número de muestras que contiene cada uno y el número de iteraciones donde se determina si existen más datos a ser leídos, calculando la diferencia entre el fragmento y el número de fragmento. En caso que fuera la última, se itera una sola vez. Con esta información se crea el fichero con extensión .H264 colocando en la cabecera los siguientes elementos:

- Un prefijo de inicio de la unidad de acceso representado por (00 00 00 01).
- La cabecera de la unidad NAL.
- E0 que representa cualquiera de los siguientes tipos de *frames (I, P, B, SI y SP).*
- Un delimitador (00 00 00 01). Esta cabecera se añade al principio de cada unidad de acceso.
- La secuencia con los datos seguidos por el delimitador (00 00 00 01).

La escritura del fichero con extensión .H264 finaliza con los datos que representan la información del vídeo. Con este proceso queda extraído el flujo elemental H.264 del contenedor MP4.

A continuación se realiza la extracción de todos los fotogramas del fichero .H264. En primer lugar se extrae el conjunto de matrices de valores [R, G, B] de cada fotograma, es decir de cada unidad de acceso del flujo elemental H.264, para posteriormente ser convertidos en imágenes con formato JPEG.

Para cada fotograma del vídeo, se extrae el residuo del ruido para posteriormente calcular la correlación del residuo del ruido entre bloques de mismo índice espacial *(i,j)* de cada par de fotogramas consecutivos entre los fotogramas extraídos del vídeo. Finalmente, se localizan bloques manipulados analizando las propiedades estadísticas de correlaciones del ruido

5.2. Diseño e Implementación del Algoritmo

Dado el coste computacional que conlleva este tipo de algoritmos de análisis estadístico, en este trabajo se ha buscado elevar la capa de abstracción, garantizando la portabilidad del código, y sin sacrificar la velocidad de ejecución. Por tanto, se ha utilizado las siguientes herramientas: Python 2.7 y OpenCV 3.1.0. La implementación se ha hecho mediante desarrollo por componentes, y utiliza las capacidades de la librería OpenCV 3.1.0. Los fotogramas y ficheros generados durante el proceso son almacenados en una carpeta temporal que se eliminan al finalizar la ejecución del algoritmo.

5.3. Evaluación del Algoritmo

Para evaluar la eficiencia del algoritmo se realizó un experimento en el que se analizó un grupo de 5 cámaras digitales de dispositivos móviles que han sido manipulados. Las marcas y modelos de los dispositivos utilizados se muestran en la Tabla 5.1.

Marca	Samsung				
Modelo	**GT-I9000**	**Galaxy A3**	**Galaxy Ace Style**	**Galaxy S5 Neo**	**Galaxy S6**

Tabla 5.1: Teléfonos móviles clasificados por marca y modelo

De cada uno de los dispositivos se recolectó un video natural, es decir, de cualquier tipo de escena sin ninguna restricción. Para este experimento se utilizó la configuración por defecto del algoritmo de control de integridad. La tabla 5.2 resume los principales parámetros utilizados.

Parámetro	Valor
Tipo de Videos	Naturales
Resolución	1920x1080
Tiempo analizado del vídeo	2 segundos
Número de fotogramas x segundo	30
Numero de Cámaras	5

Tabla 5.2: Parámetros utilizados en el experimento.

Cada video fue cortado en trozos de 2s. Como los videos son de duración de 2s, hemos obtenido un trozo único por cada video respectivamente.

De los 5 videos, con tamaño de bloque de 128x128 pixeles, 3 han mostrado suficiente porcentaje regiones poco confiables, por canal y fotograma, por lo tanto ha habido una tasa de detección del 60%

Se ha repetido el experimento, pero esta vez con bloques de 32x32 pixeles. El experimento ha tardado por lo menos diez veces más respecto del anterior, para arrojar porcentajes idénticos: 60%.

Los resultados obtenidos en el experimento se pueden observar en la Tabla 5.3

	Tamaño del Bloque	
	32x32	128x128
Tasa de Aciertos	60%	60%

Tabla 5.3: Resultados del experimento.

6. CONCLUSIONES Y TRABAJO FUTURO

6.1. Conclusiones

En este trabajo se ha desarrollado un algoritmo de control de integridad en vídeos digitales con formato MP4.

Primero, se realizó un estudio del proceso de generación de un vídeo dependiendo del tipo de dispositivo, haciendo énfasis en los tipos de codificación y la compresión que intervienen en el mismo.

Seguidamente se han comentado las principales técnicas de análisis forense orientadas a vídeos digitales, las razones que justifican su estudio. Como resultado de esta investigación y con la colaboración de expertos legales, se concluye que por sí solo un vídeo digital no es prueba suficiente para fundamentar una acusación o defensa en un proceso judicial. Por lo que es necesario contar con un segundo criterio de comprobación de integridad de un vídeo.

Posteriormente, se han revisado los principales trabajos de la literatura sobre las técnicas forenses de detección de manipulaciones en vídeos digitales. En la literatura se estudian dos tipos de manipulaciones: en los metadatos del vídeo y en los fotogramas del vídeo. Las manipulaciones en los fotogramas pueden ser resultado de una modificación en el contenido de los fotogramas o en su estructura (añadir o eliminar fotogramas).

A continuación se ha presentado la contribución de este trabajo que consiste en una técnica de control de integridad de fotogramas de vídeos de dispositivos móviles. El algoritmo fue implementado teniendo en cuenta que el coste computacional de procesamiento de imágenes es alto, por lo tanto es necesario distribuirlo entre varias máquinas o CPU's.

6.2. Trabajo Futuro

Como se ha podido observar a lo largo de todo este trabajo, el campo del análisis forense de vídeo es un tema que está en sus inicios y hay mucho que investigar y desarrollar:

- Realizar más experimentos con el algoritmo implementado para realizar mejoras y alcanzar una tasa de aciertos mayor.
- Estudiar las técnicas de detección de doble compresión de vídeos digitales.
- Implementar técnicas para detectar si un vídeo ha sido fragmentado a partir del análisis de la estructura de átomos que posee cada vídeo de una determinada marca y modelo.
- Crear herramientas que incorporen técnicas para analizar de forma integral la manipulación de un vídeo: Analizando los metadatos, contenido de audio y vídeo.

RESUMEN EN INGLÉS

7. INTRODUCTION

7.1. Motivation

Almost every modern smartphones is online at all time. We live in the age of 'Internet of things', and thus smartphones are no more 'personal computers'. By running messaging daemons, social network client software, and even malware, the smartphones, not needing human user, are able to receive, send and store pieces of information, be it pictures, video files, etc.

Imagine an user, who just found a new video file in his SmartPhone. He wonders whether he has recorded it or it has been received by an app. He wonders if a video in his phone has been tampered or not.

Nowadays, embedding metadata in video file, is not mandatory, and even avoided when a video is uploaded to a Social Network. (metadata is deleted by Social Networks). That's why, it is necessary to implement Video Forensics Analysis Techniques, in order to empower people and institutions, so they could generate classification and metadata on selected video files.

7.2. Motivation

This end of studies work (ESW) has the following objectives:

- Study existing digital video classification methods and their effectiveness.
- Choose one or several methods to check video integrity.
- Debug and Document installation and usage of the Software Development Framework at GNU/Linux, in order to get ir working on several platforms (make multi-OS and hardware independent)

7.3. Work Schedule

This project is done through four phases: Initial Phase, Documenting, Design and finally, Implementation. Additionally, each phase is made of tasks, among which are: Risk Analysis, Design, Implementation, and Testing. All of these tasks has been done simultaneously, but depending on the Phase, their importance has fluctuated.

Task	Duration (Days)	Start	End
Initial Phase	**92**	**03/10/15**	**07/02/16**
● Planning	15		
● Previous work study	30		
● Existing technologies study	15		
● Deployment and upgrade of Software Development Framework	32		
Documentation Phase	**30**	**08/02/16**	**21/03/16**
● Planning	20		
● Risk Analysis	10		
Design Phase	**30**	**22/03/16**	**28/03/16**
● Requirements Specification	10		
● Objectives Specification	200		
● Memorandum setup	15		
Implementation Phase	**40**	**10/05/16**	**04/07/16**
Testing Phase	**14**	**05/07/16**	**24/07/16**

Table 7.1. Activities of project phases

8. CONCLUSIONS AND FUTURE WORK

8.1. Conclusions

This project has issued an algorithm which checks MP4 digital video integrity.

Firstly, a study, on video creation process, has been made, through several kinds of devices, looking mostly at types of encoding and compression.

Secondly, most used techniques, on digital video forensic analysis, have been explained, and reasons to study them. As a result of this investigation, and in collaboration with legal experts, we have come to the conclusion that, solely a video file, only by itself, is not enough evidence to support a legal defense or an prosecution.

Thirdly, we have read the leading studies about forensic techniques on digital video tampering detection. In these studies, it's written about two areas of tampering: metadata versus video photograms.

Lastly, the contribution issued by this project, is made of a technique that checks integrity of mobile device video photograms. The algorithm has been implemented in consideration of image processing's high computational cost. That's why, it can be (but not necessarily) distributed over several machines or CPU Cores.

8.2. Future Work

As you may have seen, by reading about this project, Video Forensic Analysis Field is in early stages, and there's still a lot to be implemented and tested:

- Make further testing and improving , using implemented algorithm, in order to achieve bigger success rate
- Study double compression detection techniques
- Implement techniques oriented to detecting whether a video has been fragmented. That is done based on analysis of atoms' structure typical of certain brand and models.
- Create all-in-one tools made with techniques that analyze metadata, video and audio streams and others.

REFERENCIAS

[AZ06] M. Al-Zarouni. Mobile Handset Forensic Evidence: a Challenge for Law Enforcement. In *Proceedings of the 4th Australian Digital Forensics Conference*. School of Computer and Information Science, Edith Cowan University, December 2006.

[BFM+12a] P. Bestagini, M. Fontani, S. Milani, M. Barni, A. Piva, M. Tagliasacchi, and S. Tubaro. An Overview on Video Forensics. In *APSIPA Transactions on Signal and Information Processing*, volume 1, pages 1229–1233. APSIPA, August 2012.

[BFM+12b] P. Bestagini, M. Fontani, S. Milani, M. Barni, A. Piva, M. Tagliasacchi, and S. Tubaro. An Overview on Video Forensics. In *APSIPA Transactions on Signal and Information Processing*, volume 1, pages 1229–1233. APSIPA, August 2012.

[BSM08] S. Bayram, H. T. Sencar, and N. Memon. Classification of Digital Camera-Models Based on Demosaicing Artifacts. *The International Journal of Digital Forensics & Incident Response*, 5(2):49–59, September 2008.

[C.B05] A. C.Bovik. *Handbook of Image and Video Processing (Communications, Networking and Multimedia)*. Academic Press, Inc., Orlando, FL, USA, 2005.

[CGS98] G Ciocca, I Gagliardi, and R Schettini. Retrieving Color Images by Content. In *Proceedings of the Image and Video Content-Based Retrieval Workshop*, 1998.

[Cor12] Panasonic Corporation. Lumix Digital Camera Know-Hows, 2012.

[HHLH08] C.C. Hsu, T. Y. Hung, C. W. Lin, and C.T Hsu. Video forgery detection using correlation of noise residue. In *Multimedia Signal Processing, 2008 IEEE 10th Workshop*, pages 170–174, October 2008.

[HJSW16] P. He, X. Jiang, T. Sun, and S. Wang. Double compression detection based on local motion vector field analysis in static-background videos. *Visual Communication and Image Representation*, 35(C):55–66, February 2016.

[Iai10] E. R. Iain. *The H.264 Advanced Video Compression Standard*. Wiley, London, UK, 2010.

[Inc12] Texas Instruments Incorporated. Digital Still Camera, 2012.

[Int96] International Organization for Standardization. Information technology - Generic Coding of Moving Pictures and Associated Audio Information: Video, 1996.

[Int16] International Telecomunication Union. H.264 : Advanced Video Coding for Generic Audiovisual Services, 2016.

[JWS+13] X. Jiang, W. Wang, T. Sun, Y. Q. Shi, and S. Wang. Detection of Double Compression in MPEG-4 Videos Based on Markov Statistics. *IEEE Signal Processing Letters*, 20(5):447–450, May 2013.

[LBR+13] D. Labartino, T. Bianchi, A. De Rosa, M. Fontani, D. Vázquez-Padín, A. Piva, and M. Barni. Localization of Forgeries in MPEG-2 Video Trough GOP Size and DQ Analysis. In *Multimedia Signal Processing (MMSP), 2013 IEEE 15th International Workshop on*, pages 494–499, September 2013.

[Li10] C. T. Li. Source Camera Identification Using Enhanced Sensor Pattern Noise. *IEEE Transactions on Information Forensics and Security*, 5(2):280–287, June 2010.

[LWH08] W. Luo, M. Wu, and J. Huang. MPEG Recompression Detection based on Block Artifacts. In *Proceedings of the International Society for Optics and Photonics Electronic Imaging*, volume 6819, pages 68190X–68190X–12, 2008.

[Moe12] T. B. Moeslund. *Introduction to Video and Image Processing, Building Real Systems and Applications*. Springer-Verlag, London, August 2012.

[MPFL] J.L Mitchell., W.B. Pennebaker, C.E Fogg, and D.J Legall. *MPEG Video Compression Standard*. Chapman and Hall, London, UK, UK.

[MTT12] S. Milani, M. Tagliasacchi, and S. Tubaro. Discriminating Multiple JPEG Compression Using First Digit Features. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2253–2256, March 2012.

[Nak05] J. Nakamura. *Image Sensors and Signal Processing for Digital Still Cameras*. CRC Press, Boca Raton, FL, USA, August 2005.

[RP07] A. R. Reibman and D. Poole. Characterizing Packet-Loss Impairments in Compressed Video. In *IEEE International Conference on Image Processing*, volume 5, pages 77–80, September 2007.

[SLL12] M. C. Stamm, W. S. Lin, and K. J. R. Liu. Temporal Forensics and Anti-Forensics for Motion Compensated Video. *IEEE Transactions on Information Forensics and Security*, 7(4):1315–1329, August 2012.

[SSCL15] P.C. Su, P. L. Suei, M.K Chang, and J. Lain. Forensic and Anti-forensic Techniques for Video Shot Editing in H.264/AVC. *Journal of Visual Communication and Image Representation*, 29(C):103–113, may 2015.

[SWJ12] T. Sun, W. Wang, and X. Jiang. Exposing Video Forgeries by Detecting MPEG Double Compression. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1389–1392, March 2012.

[THN+] M. Tennoe, E. Helgedagsrud, M. Naess, H. Kjus Alstad, H. Kvale Stensland, V. Gaddam, Reddy Gaddam, D. Johansen, C. Griwodz, and P Halvorsen.

[VPFB+12] D. Vazquez-Padin, M. Fontani, T. Bianchi, P. Comesana, A. Piva, and M. Barni. Detection of Video Double Encoding with GOP Size Estimation. In *IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 151–156, December 2012.

[WBSH09] A. Wahab, J.A. Briffa, H. G. Schaathun, and A. T. S. Ho. Conditional Probability Based Steganalysis for JPEG Steganography. In *Proceedings of the International Conference on Signal Processing Systems*, pages 205–209, Singapore, May 2009.

[WF06] W. Wang and H. Farid. Exposing Digital Forgeries in Video by Detecting Double MPEG Compression. In *Proceedings of the 8th Workshop on Multimedia and Security*, MM&Sec '06, pages 37–47, New York, NY, USA, 2006. ACM.

[WHL12] A. Wahab, A. T. S. Ho, and S. Li. Inter-Camera Model Image Source Identification with Conditional Probability Features. In *Proceedings of the IIEEJ 3rd Image Electronics and Visual Computing Workshop*, Kuching, Malaysia, November 2012.

[Woo05] C. Wootton. *A Practical Guide to Video and Audio Compression*. Elsevier, Burlington, USA, January 2005.