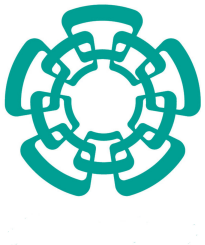

Centro de Investigación y de Estudios Avanzados
del Instituto Politécnico Nacional

Unidad Cinvestav Tamaulipas

# Método de rastreo de copias de video comprimido en formato H.264/AVC resistente a ataques de colusión

Tesis que presenta:

**Raúl Quiñones Arteaga**

Para obtener el grado de:

**Maestro en Ciencias en Ingeniería y Tecnologías Computacionales**

Director de la Tesis:
Dr. José Juan García Hernández

Cd. Victoria, Tamaulipas, México. Septiembre, 2017

La tesis presentada por Raúl Quiñones Arteaga fue aprobada por:

_____

Dr. Miguel Morales Sandoval

Dr. Hiram Galeana Zapién

Dr. José Juan García Hernández, Director

Cd. Victoria, Tamaulipas, México., 29 de Septiembre de 2017

A toda mi familia, en especial a mis papás Raúl e Imelda y mis hermanos Sergio, Hilda y Toño.

# Agradecimientos


- A Dios por darme la oportunidad de cumplir una meta más en mi vida.
- A mis padres por todo lo que me han enseñado, sobre todo por el apoyo brindado en todas y cada una de las etapas de mi vida. A mis hermanos por el apoyo incondicional y cariño que siempre me han demostrado.
- A mi tía Chela y mi prima Alba por sus muestras de cariño y por todo el apoyo brindado.
- A mi novia Fátima por siempre darme los ánimos necesarios para seguir adelante en este objetivo.
- Al Dr. José Juan García Hernández, por su confianza, su paciencia y el apoyo al guiarme en todo momento para realizar este trabajo de investigación.
- A todos los investigadores por compartirme sus conocimientos, especialmente a los Drs. Hiram Galeana Zapién y Miguel Morales Sandoval por sus acertadas observaciones y comentarios en el trabajo investigación realizado.
- Al Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional (CINVESTAV-Tamaulipas) por la oportunidad de realizar la maestría.
- A todo el personal que elabora en el CINVESTAV-Tamaulipas por su apoyo y dedicación.
- Al Consejo Nacional de Ciencia y Tecnología (CONACyT) por el apoyo económico proporcionado durante mi instancia de la maestría el cual me permitió dedicarme exclusivamente a mis estudios y la investigación realizada.

# Índice General

# Índice de Figuras

# Índice de Tablas

# Resumen

## Método de rastreo de copias de video comprimido en formato H.264/AVC resistente a ataques de colusión

por

**Raúl Quiñones Arteaga**
Unidad Cinvestav Tamaulipas
Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, 2017
Dr. José Juan García Hernández, Director

Las herramientas tecnológicas actuales permiten que la edición y transferencia de archivos multimedia entre usuarios, como el audio y video, sea un proceso fácil y muy frecuente. Estas mismas facilidades, sin embargo, han propiciado que la piratería de archivos multimedia crezca. Entre los archivos multimedia que sufren más piratería se encuentra el video. Para enfrentar este problema se han desarrollado métodos que permiten identificar a las personas que distribuyen los archivos de video de manera ilícita.

El *fingerprinting* es una técnica que permite rastrear el origen del video que ha sido distribuido de manera ilegal. La forma de detectar al usuario que distribuye el video de una forma no autorizada es mediante la inserción al video de un identificador no perceptible al ojo humano. Típicamente, esta inserción se hace con técnicas de *marcado de agua digital*. Sin embargo, uno o varios usuarios deshonestos, también llamados traidores, pueden usar métodos para tratar de borrar o alterar los identificadores y de esta forma evitar que sea descubierto el propietario del archivo original. Uno de los métodos utilizados para dicho fin son los ataques de colusión.

Por otro lado, el formato de compresión H.264/AVC es en la actualidad el más utilizado para la distribución de video de alta definición debido a sus altas tasas de compresión. El estándar H.264/AVC utiliza una compresión con pérdida mediante el uso de la transformada DCT entera y una compresión sin pérdida utilizando una codificación entrópica. En este trabajo se presenta un método de rastreo de copias de video comprimido en H.264/AVC robusto a ataques de colusión utilizando códigos identificadores con un enfoque de agrupamiento. Los resultados obtenidos son competitivos a los

presentados en el estado del arte ya que el número de usuarios coludidos entre sí detectados es igual o mayor a los que reportan usando otras propuestas. Además, se comprobó que el uso de un enfoque de agrupamiento permite hacer uso de códigos de longitud mínima. El uso de estos códigos se ve reflejado en el tiempo de video necesario para realizar la detección.

# Abstract

## Method for tracking H.264/AVC compressed video copy resistant to collusion attacks

by

**Raúl Quiñones Arteaga**
Cinvestav Tamaulipas
Center for Research and Advanced Studies of the National Polytechnic Institute, 2017
Dr. José Juan García Hernández, Advisor

The current technology tools allow the easy edition and transfer of multimedia files, as audio and video, between users. These same facilities, however, have encouraged that the multimedia file piracy increases. The video is one of the most pirated multimedia files. Have been developed methods that allow identify the people who distribute video files illicitly to confront the piracy problem.

Fingerprinting is a technique that allows you to trace the origin video that has been illegally distributed. The way to track the user who distributes the video in an unauthorized way is by inserting a non-perceptible identifier for the human eye. Commonly, this insertion is done with digital watermarking techniques. However, one or more dishonest users, also called traitors, may use methods to try to erase or alter the identifiers and thus prevent the owner of the original file from being discovered. One of the methods used for this purpose is know as collusion attacks.

On the other hand, the H.264/AVC compression format is currently the most used for the distribution of high definition video due to its high compression rates. H.264/AVC standard uses loss compression by using integer DCT transform and lossless compression using entropic coding. In this work are present a method for tracing H.264/AVC compressed video copies using identifier codes with a grouping approach. The results obtained are competitive to the presented in the state of the art since the number of colluded users detected is equal or greater than the reported using other proposals. In addition, it was found that the use of a grouping approach allows the use of minimum length codes. The use of these codes is reflected in the video time required to perform the detection.

# Nomenclatura

| | |
|---|---|
| **MPEG** | Moving Picture Experts Group |
| **VCEG** | Video Coding Experts Group |
| **JPEG** | Joint Photographic Experts Group |
| **DCT** | Discrete Cosine Transform |
| **IDCT** | Inverse Discrete Cosine Transform |
| **QP** | Quality Parameters |
| **CDMA** | Code Division Multiple Access |
| **JND** | Just Noticeable Distortion |
| **VCL** | Video Coding Layer |
| **CAVLC** | Context-Adaptive Variable Length Coding |
| **CABAC** | Context-adaptive binary arithmetic coding |
| **GOP** | Group of Pictures |
| **PSNR** | Peak Signal to Noise Ratio |
| **MSE** | Mean Squared Error |
| **ACC** | Anti-Collusion Code |
| **QR** | Quick Response Code |
| **RMS** | Root Mean Square |
| **ECC** | Error Correction Code |
| **PSE** | Permuted Subsequent Embedding |
| **BTC** | Block Truncation Coding |
| **DWT** | Discrete Wavelet Transform |
| **SVD** | Singular Value Decomposition |

# 1

# Introducción

En este capítulo de manera breve se describe los antecedentes y motivaciones que llevaron a realizar este trabajo de tesis, de igual forma, se describen los objetivos planteados y la metodología a seguir para el desarrollo de este trabajo.

## 1.1 Antecedentes y motivación

El avance tecnológico de herramientas multimedia que existe hoy en día permite que la transferencia y la edición de archivos multimedia sean tareas fáciles y muy frecuentes. Estas facilidades han permitido que la piratería de archivos multimedia crezca de manera considerable. Entre los archivos multimedia que sufren más plagio se encuentran los archivos de audio (música) y los de videos (películas) [11]. En el caso particular de los archivos de video, la venta y la distribución de películas piratas es la principal actividad ilícita que se realiza. Existen diferentes herramientas que permiten compartir archivos multimedia por medio de Internet de una manera fácil, esto ha llevado a que la piratería de archivos en línea, principalmente música y películas, sea muy recurrente.

Los archivos de música al ser de un tamaño pequeño suelen obtenerse desde una descarga directa, mientras que para los archivos de películas suelen utilizarse diferentes protocolos *peer-to-peer* como *BitTorrent* desarrollado en 2003 y que permite compartir dichos archivos de una manera más rápida debido a que la descarga se hace simultáneamente desde diferentes servidores [11]. La distribución de piratería de películas ocupa una de las principales actividades en Internet, aproximadamente el 30 % del ancho de banda del Internet es usado por usuarios para el intercambio de archivos multimedia donde predominan los archivos de video [58]. Estudios acerca de la piratería indican que existen pérdidas de ingresos a causa de la piratería en línea de hasta el 40 % reportadas por propietarios de derechos de autor [56]. Otros estudios revelan que la piratería de películas en el año 2012 provocó que 72,000 personas perdieran su empleo, así como $4.5 billones de pérdidas en salarios en la economía de los Estados Unidos [47].

Diferentes medidas de prevención como la conciencia social se han llevado a la práctica pero lamentablemente no han obtenido resultados favorables [11]. Para contrarrestar la piratería de videos actualmente en la literatura se han presentado esquemas, utilizando técnicas *watermarking*, *cifrado* y *fingerprinting* [36], que permitan identificar al dueño del video original del cual se han obtenido las copias ilegales. Teniendo en cuenta la diferencia entre *marcado de agua*, *cifrado* y *fingerprinting*, como se explica en [46], el marcado de agua se usa para la protección de derechos, *cifrado* es usado para restringir el acceso al contenido de los archivos multimedia a personas no autorizadas y *fingerprinting* es la aplicación usada para el rastreo de usuarios deshonestos.

Dicho rastreo se realiza almacenando información de derechos de autor en un archivo multimedia es utilizando una técnica de marcado de agua. Tal como se muestra en la Figura 1.1 en el proceso de marca de agua la información, conocida como marca de agua, puede presentarse tanto de manera visible como invisible. Con dicha marca se puede reclamar la pertenencia de un archivo multimedia. Comúnmente la marca se inserta de manera invisible para que no sea detectable por el usuario y no

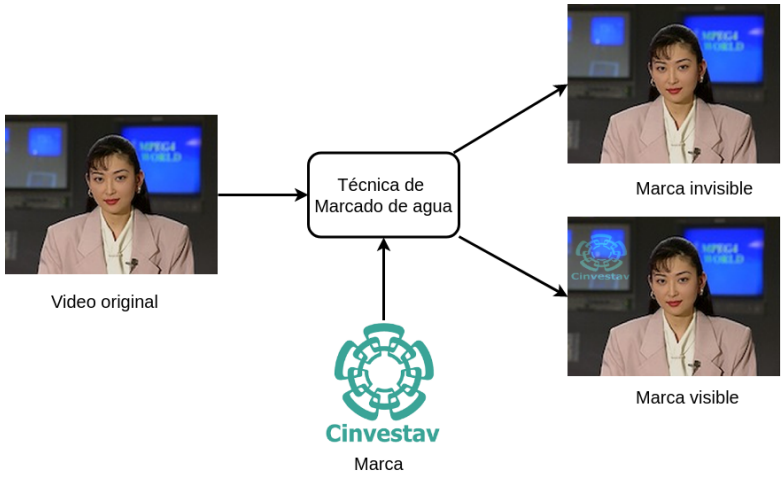


Figura 1.1: Proceso de inserción de un identificador con marcado de agua.

se elimine intencionalmente.

El *fingerprinting*, término introducido por Wagner [51] en 1983, es el proceso de inserción de un conjunto de marcas a una señal anfitrión para producir un conjunto de firmas digitales. Este método en combinación con una técnica de marcado de agua permite rastrear el origen del video distribuido de manera ilegal. Esta técnica permite que la marca sea imperceptible para el usuario, por lo que si un usuario deshonesto es partícipe del proceso ilícito de distribución de películas se tendrán argumentos suficientes para identificarlo y proceder con las sanciones correspondientes.

Aún con las medidas tomadas por distribuidoras de películas para detener la piratería, existen personas insistentes en realizar actos de piratería por lo que usan métodos para eliminar o distorsionar la información y de esta forma evitar que sea descubierto el propietario del video original. Uno de los métodos utilizados es el ataque de colusión [7]. Existen diferentes formas de efectuar este ataque, uno de los más usados es el promedio en donde un grupo de usuarios unen sus videos originales y calculan un promedio de dichos videos como se ilustra en Figura 1.2, en el video resultante se conoce como video pirata. Con este tipo de ataque, la información que comparten en común cada uno de los videos se mantiene similar, siendo el identificador del usuario en cada video el que es afectado y por lo tanto no hay información para detectar al usuario original.

Para generar robustez contra estos ataques se hace uso de códigos resistentes a colusión [59],

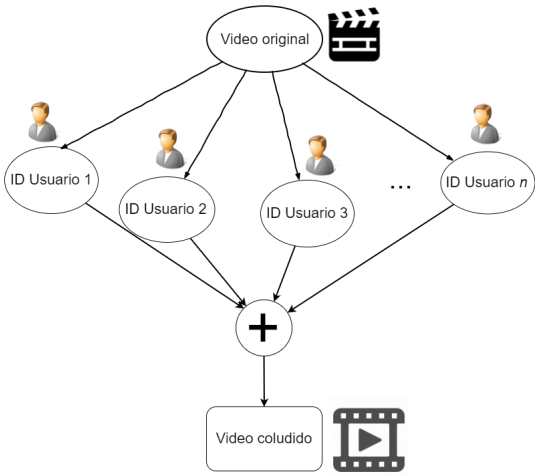


Figura 1.2: Ataque de colusión.

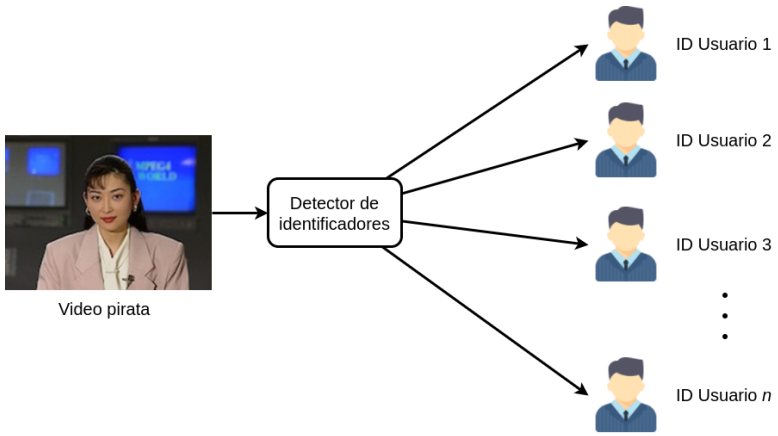


Figura 1.3: Proceso detección con marcado de agua.

dichos códigos deben ser capaces de mantenerse aún después del ataque. Como se muestra en la Figura 1.3 se pueden obtener varios identificadores a partir del video pirata, dichos identificadores pertenecen a cada uno de los usuarios que participaron en el ataque.

Por otro lado, H.264/AVC es un estándar de codificación de video de ITU-T e ISO/IC el cual ofrece mejor relación compresión-calidad comparada con estándares de videos anteriores como lo son MPEG-2 y MPEG-2 parte 2 [13]. Existen varias características que diferencian al H.264/AVC de los anteriores estándares entre las que destacan el hecho de que pueda trabajar con bloques de 4x4

píxeles mientras los estándares predecesores trabajan con bloques de $8 \times 8$. La transformada de coseno discreta (DCT, por sus siglas en Inglés), que utilizan compresores anteriores, es sustituida por la transformada entera la cual se puede calcular con sumas y corrimientos evitando las multiplicaciones, por lo cual el cálculo de la transformada se vuelve computacionalmente más rápido. El H.264/AVC al ser uno de los estándares de codificación que proporciona mayores beneficios de compresión, es utilizado en la transmisión de televisión digital terrestre, por cable y por satélite, además de ser usado en el formato *Blu-ray*, así como en servicios de video streaming [55]. Por estas razones el estándar H.264/AVC es empleado para comprimir las películas de video digital. La compañía de servicios multimedia en la nube *encoding.com* reporta que el formato de codificación H.264/AVC es el más utilizado por sus usuarios con el 72 % [14].

Enfrentar el problema de la distribución ilegal de películas es de gran interés para las industrias cinematográficas ya que permitiría detectar todo aquel usuario traidor y de esta forma contrarrestar las tasas de piratería. Esto último permitiría reducir las pérdidas monetarias y por ende se mantendrían y generarían más empleos.

## 1.2 Planteamiento del problema

Los usuarios que adquieren los archivos de películas de una forma lícita tienen acceso total a dicho archivo, por lo que usuarios deshonestos aprovechan esta ventaja para promover la piratería. Esto ha ocasionado que en el escenario de distribución de películas sea necesario el uso de una herramienta de seguridad, que si bien no impida la distribución ilegal de las películas, permita rastrear a los usuarios que adquirieron los archivos originales. Las técnicas *fingerprinting* son usadas para dicho fin [4].

En los últimos años se han propuesto esquemas *fingerprinting* basados en teorías de codificación, los códigos más representativos son los códigos de Tardos los cuales representan una detección analítica. Sin embargo, los códigos de Tardos suelen ser de longitud extensa debido a que crece

cuadráticamente con respecto al número de usuarios coludidos entre sí. Debido a ello hace que no sea viable en un escenario real ya que es necesario considerar un gran número de usuarios y por ende la longitud del código sería demasiado extensa, esto hace necesario explorar alternativas con identificadores más cortos con una eficiencia igual o mejor.

Por otra parte, el concepto de agrupamiento de usuarios, propuesto en [54] para la creación de códigos resistentes a colusión, ha demostrado ser eficiente tanto en documentos digitales [38] como en archivos de audio [15], en donde los usuarios son agrupados con base en sus características similares como puede ser el área geográfica.

Teniendo en cuenta que el *fingerprinting* permite el rastreo de usuarios coludidos entre sí, que el formato de codificación H.264/AVC es altamente usado por sus grandes ventajas y que el enfoque de agrupamiento de usuarios permite usar códigos identificadores más cortos, se plantea la siguiente pregunta de investigación:

**¿Es posible rastrear copias de video codificadas en formato H.264/AVC con un enfoque robusto a ataques de colusión a través de códigos de longitudes mínimas mediante el concepto de agrupación de usuarios?**

## 1.3 Objetivos

### 1.3.1 Objetivo general

Desarrollar un método para el rastreo de copias de video codificadas en el formato H.264/AVC robusto a ataques de colusión utilizando códigos de longitud mínima y agrupamiento de usuarios.

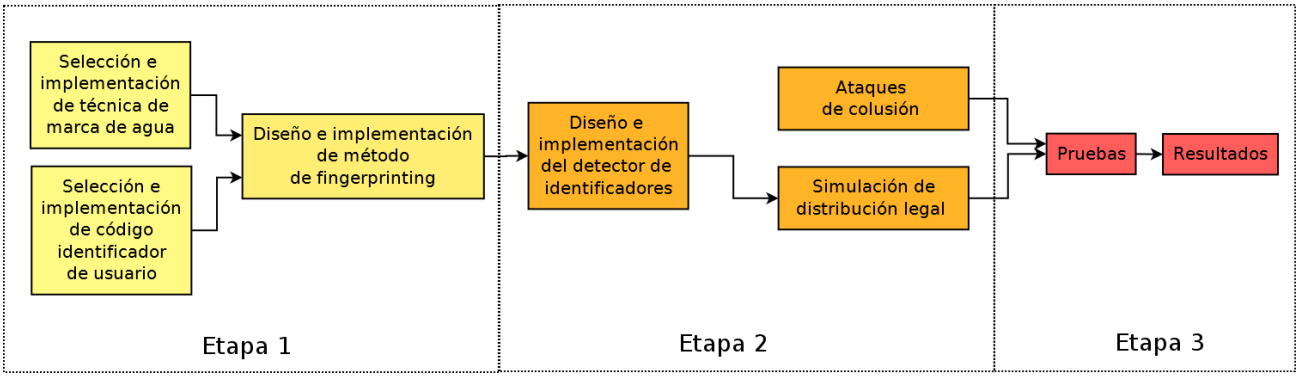


Figura 1.4: Etapas de la metodología.

### 1.3.2 Objetivos particulares

- Determinar e implementar un generador de código apropiado que identifique a cada usuario, que sea robusto a ataques de colusión y sea de menor longitud que los códigos de Tardos.
- Seleccionar e implementar la técnica de marcado de agua adecuada para el formato de video H.264/AVC que permita insertar el código de manera transparente.
- Diseñar e implementar el detector de usuarios adecuado que permita maximizar el número de usuarios coludidos entre sí y determinar el tiempo mínimo de video necesario para realizar dicha detección.

## 1.4 Metodología

En esta sección se describe la metodología que se siguió para alcanzar los objetivos planteados. La metodología se divide en tres etapas, las cuales se pueden observar en la Figura 1.4. Cada una de las etapas se describen a continuación:

- Etapa 1: Selección e implementación del generador de identificadores.

  En esta etapa se estudia el estándar de compresión H.264/AVC para conocer sus características particulares. También, se obtiene y se analiza la herramienta JM (Joint Test Model) que es el

software de referencia del H.264/AVC. Una vez conocidas las características del H.264/AVC, se revisa las técnicas de *fingerprinting* reportadas en trabajos identificando las marcas y técnicas de marcado de agua empleadas, posteriormente se selecciona el identificador a utilizar y la técnica de marcado de agua más apropiada a las características del H.264/AVC. Finalmente, se realiza la implementación de un generador de identificadores así como la técnica de marca de agua seleccionada.

- Etapa 2: Diseño del detector de usuarios.

  Ya que se tiene la implementación para generar e insertar un identificador a cada video, se diseña un detector e identificador de usuarios a partir de un modelo probabilístico del comportamiento de los identificadores del video pirata determinado mediante experimentación. El modelo es implementado teniendo así un sistema de generación, inserción y detección de identificadores en videos. Este sistema permite generar una simulación de distribución legal y a partir de los videos que han sido "distribuidos" se generan una serie de ataques de colusión con características particulares.

- Etapa 3: Pruebas y resultados.

  Finalmente, en esta etapa se pone a prueba el método propuesto evaluando cada uno de los resultados obtenidos en cada ataque generado, dichos resultados son comparados con los reportados en el estado del arte y se determina si son competitivos destacando las cualidades que presenta el método propuesto.

## 1.5 Organización del documento

El resto del documento se organiza de la siguiente manera. En el Capítulo 2 se describen la definición, las características, clasificación y algunas aplicaciones del marcado de agua. En el Capítulo 3 se describen las técnicas de marcado de agua así como los identificadores utilizados en diferentes

trabajos en video que se encuentran en la literatura poniendo énfasis en aquellos que son resistentes a ataques de colusión. En el Capítulo 4 se describe a detalle la metodología que se realizó para desarrollar el método de rastreo para copias de video. Por otra parte, en el Capítulo 5 se describen las pruebas realizadas así como los respectivos resultados obtenidos. Finalmente, en el Capítulo 6 se presentan las conclusiones a las que se llegaron con los resultados obtenidos y el trabajo a futuro.

# 2

# Marco teórico

Para conseguir el objetivo propuesto en este trabajo es necesario conocer el estándar H.264/AVC, el tipo de técnicas de marcado de agua, así como la composición de los códigos resistentes a ataques de colusión. Por ello, en este capítulo se describen las características particulares del estándar H.264/AVC así como el proceso de codificación/decodificación que se lleva a cabo en dicho estándar. También, se muestra la clasificación de las técnicas de *marcado de agua* observando sus ventajas y desventaja. De igual forma se presentan algunos de los códigos *fingerprinting* que son propuestos en la literatura.

## 2.1 Estándar H.264/AVC

**H.264/AVC (Advanced Video Coding)** o **MPEG-4 parte 10** es un estándar de codificación desarrollando por la ITU-T Video Coding Expert Group (VCEG) y la ISO/IEC Moving Picture Experts Group (MPEG) creado para mejorar las características de los estándares predecesores MPEG-2, H.263 o MPEG-4 parte 2 debido a la proliferación de video digital dentro de los nuevos espacios de aplicación

tales como televisión móvil o televisión en alta definición. Algunas de las características con las que cuenta el formato H.264/AVC y que lo diferencian de los estándares predecesores son [24]:

- **Bloques de** $4 \times 4$.- Trabaja con bloques de $4 \times 4$ mientras que en estándares anteriores trabajan con bloques mínimos de $8 \times 8$.
- **Transformada entera**.- Se sustituye la DCT por la transformada entera, la cual se calcula con un costo computacional menor al utilizar sólo sumas y desplazamientos.
- **Compensación de movimiento de cuarto de precisión**.- Provee una exactitud de estimación de movimiento hasta de un cuarto de píxel comparado con la estimación de medio píxel que presentan estándares anteriores.
- **Filtro de deblocking**.- Usa un filtro antibloques que mejora la eficiencia de compresión y calidad visual.

La capa de codificación de video (VCL, por sus siglas en Inglés) de H.264/AVC usa el mismo principio que los otros estándares, como MPEG-2, en el cual se utiliza un enfoque híbrido de predicción *temporal* y *espacial* basado en bloques en conjunto con una codificación de transformada basada en bloques [35]. En la Figura 2.1 se muestra el proceso de codificación y decodificación del estándar H.264/AVC. Cada *frame* del video es dividido en bloques denominados *macrobloques*, los cuales son bloques de $16 \times 16$ píxeles de componente *luminancia* y, para el caso del formato de muestreo crominancia 4:2:0, $8 \times 8$ píxeles de cada uno de los dos componentes *crominancia*. Cada macrobloque por separado pasa por el proceso de codificación/decodificación que se muestra en la Figura 2.1.

### 2.1.1 Proceso de codificación

Cada uno de los procesos que se realizan en la compresión de video se describen a continuación:

1. **Predicción**.- Se calcula un macrobloque mediante una predicción (las diferentes predicciones se discuten en la Sección 2.1.3) del macrobloque actual utilizando otros píxeles, del mismo

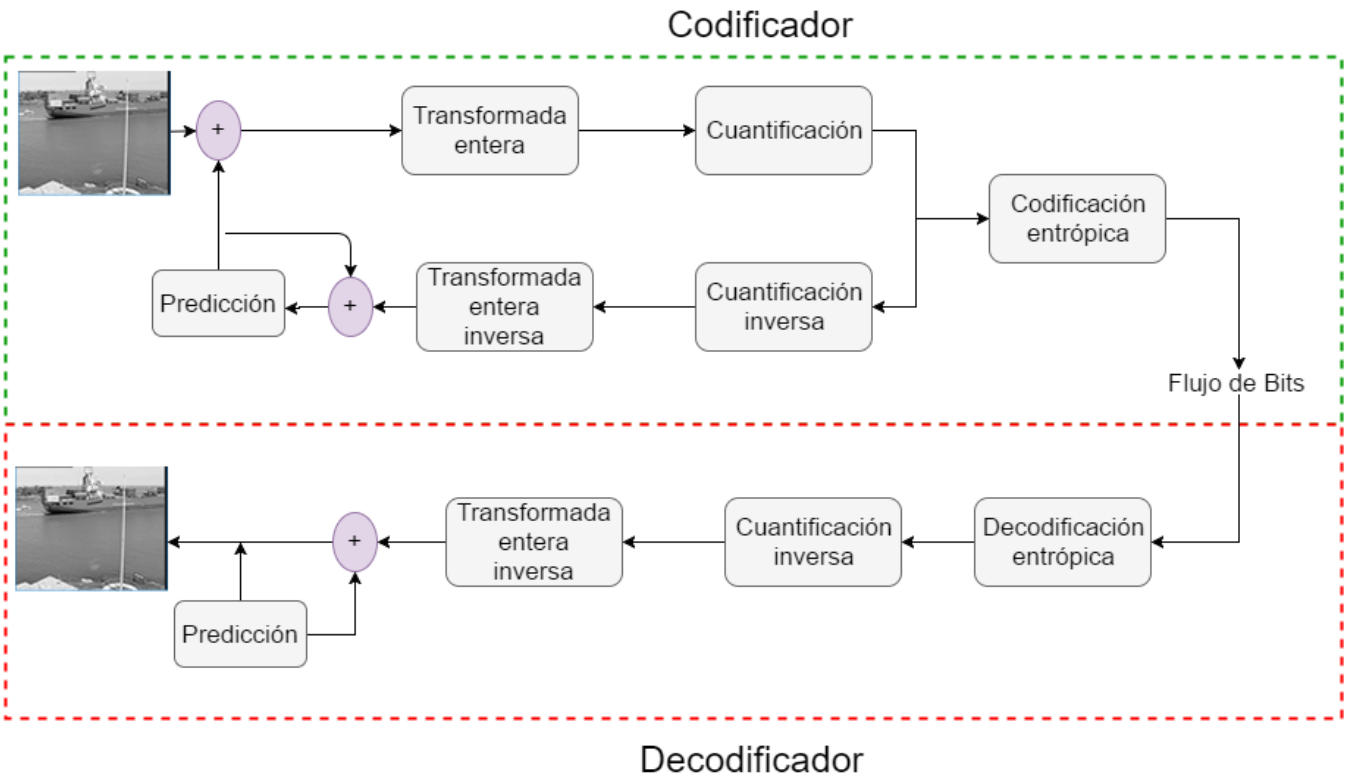


Figura 2.1: Proceso de codificación y decodificación del estándar H.264/AVC [40].

u otro *frame*, como referencia. La diferencia entre el macrobloque original y el macrobloque predicho genera un error (residuo) el cual continúa a los siguientes bloques del proceso de codificación. Existen dos tipos de predicción según su referencia.

- Predicción Intra.- La predicción del macrobloque se genera a partir de los píxeles vecinos del mismo macrobloque.
- Predicción Inter.-La predicción del macrobloque se genera a partir de los píxeles de macrobloques que pertenecen a diferente *frame*.

En el estándar H.264/AVC existen tres tipos de *frames*: *I*, *P* y *B*. El tipo de *frame* se define dependiendo de la predicción que se utilice en los macrobloques que conforman dicho *frame*. Los *frames I* utilizan sólo predicción *Intra*, los *frames P* predicción *Inter* con referencia a sólo *frames* anteriores y los *frames B* predicción *Inter* con referencia a *frames* anteriores y posteriores. En la Figura 2.2 se observan los tipos de *frames* en el estándar H.264/AVC.

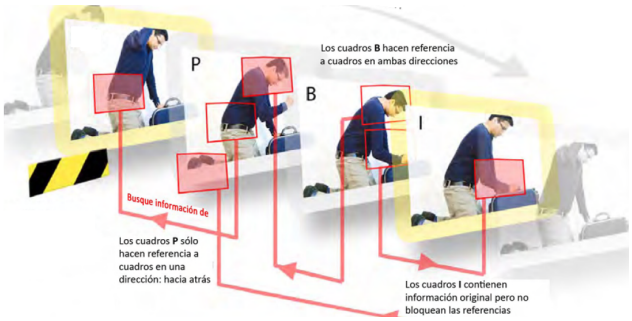


Figura 2.2: Tipos de *frames* en el estándar H.264/AVC [30].

2. **Transformada entera**.- Al bloque de residuo se le realiza una transformación del dominio del espacio al dominio de la frecuencia utilizando *la transformada entera* la cual se calcula con sumas y desplazamientos por lo que es un proceso computacionalmente ágil. Dicha transformada es una aproximación de la DCT. En la Figura 2.3 se ilustra el resultado de una

DCT entera, en donde se puede observar los coeficientes pertenecientes a las bajas, medias y altas frecuencias, además, se enumera el recorrido *zigzag* con el cual se toman los coeficientes.

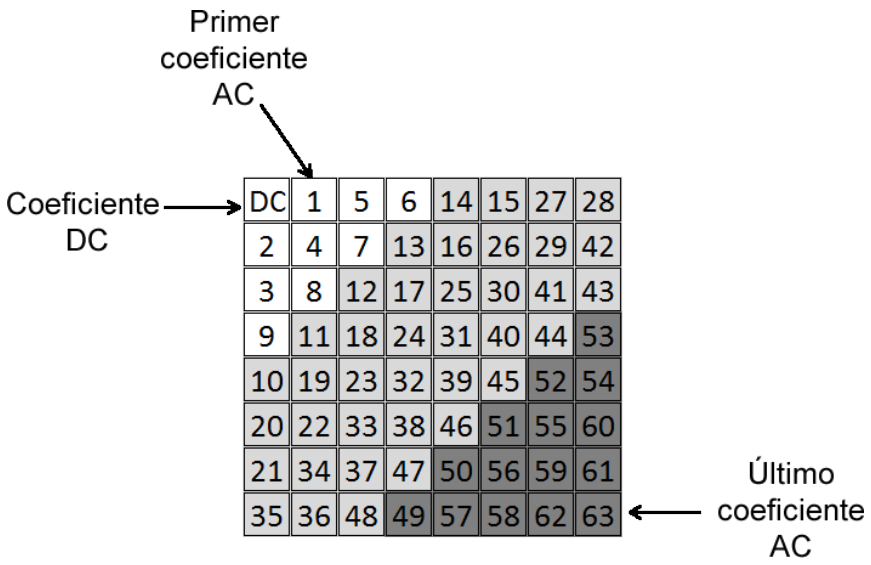


Figura 2.3: Coeficientes de una DCT entera.

3. **Cuantificación**.- Cada uno de los coeficientes resultantes de la transformada es dividido por una valor entero y redondeado al entero más cercano con el fin de reducir el valor de los coeficientes, algunos hasta cero, como se muestra en la Figura 2.4. La cuantificación reduce la precisión de los coeficientes de la transformada de acuerdo a un *parámetro de cuantificación* (QP). Mientras más grande sea el valor QP mayor será el número de coeficientes cero y por lo tanto mayor la compresión a costa de una imagen decodificada de calidad pobre. Por lo contrario, si el valor QP es bajo habrá más coeficientes no cero y por lo tanto una imagen decodificada de mayor calidad pero con una compresión muy baja.

4. **Codificación entrópica**.- En este paso se crea el flujo de bits mediante un proceso de compresión (sin pérdida) de toda la información generada. Entre la información a codificar se encuentran valores y parámetros como:

   - Coeficientes cuantificados.

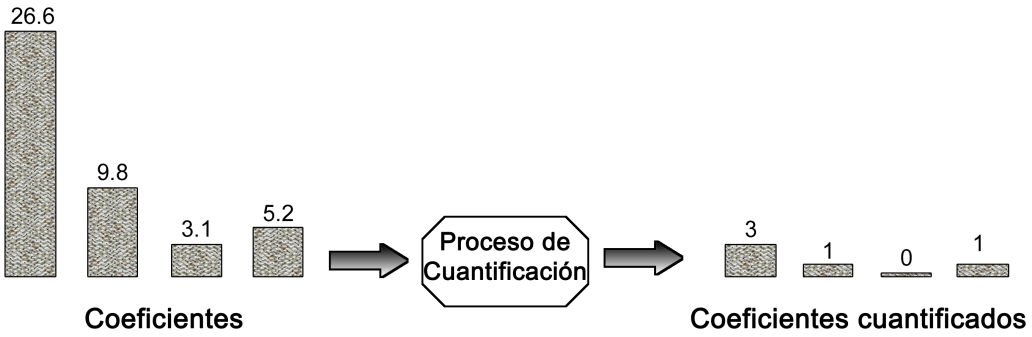


Figura 2.4: Proceso de cuantificación de los coeficientes.

- Información que permite al decodificador recrear la predicción .
- Información acerca de la estructura de los datos y las herramientas utilizadas durante la compresión.
- Información acerca de la secuencia del video completo.

Estos valores y parámetros son convertidos en códigos binarios usando dos posibles técnicas de compresión sin perdida [23], las cuales son:

- **Adaptación a contexto de codificación de longitud variable** (CAVLC, por sus siglas en inglés), es una variante adaptativa de la codificación Huffman dirigido a aplicaciones que requieren un decodificador de entropía ligeramente más simple.
- **Adaptación a contexto de codificación aritmética binaria** (CABAC, por sus siglas en inglés), el cual está basado en técnicas de codificación aritméticas. La diferencia de la codificación aritmética con respecto a la codificación Huffman es que no codifica cada símbolo por separado, sino que codifica un mensaje completo en un solo número entre 0.0 y 1.0.

### 2.1.2 Proceso de decodificación

El proceso de decodificación hace uso de los mismos procesos del codificador pero en forma inversa con el siguiente orden.

1. **Decodificación entrópica**.- El decodificador recibe el flujo de datos, éste es decodificado para obtener los coeficientes cuantificados y la información que permitirá recrear la secuencia de los frames.

2. **Cuantificación inversa**.- Cada coeficiente es multiplicado por un entero dado por el parámetro de cuantificación QP, el cual es el mismo utilizado en la compresión. El resultado obtenido será una aproximación a los coeficientes originales, la precisión del valor estará dada por el valor QP que se haya usado.

3. **Transformada entera inversa**.- Los coeficientes descuantificados se transforman del dominio frecuencial al espacial mediante la transformada entera inversa y se obtiene el bloque de residuo.

4. **Predicción**.- Finalmente, a partir de bloques de residuo y junto con los bloques de referencia se obtienen la aproximación de los bloques originales de cada uno de los *frames* de video.

### 2.1.3 Predicción

#### *2.1.3.1. Predicción Intra*

La predicción *Intra* de un *frame* se realiza mediante la predicción de cada uno de sus macrobloques. Para cada predicción del macrobloque se puede realizar una subdivisión $4 \times 4$, $8 \times 8$ o $16 \times 16$; según el tamaño del subbloque existen diferentes modos para calcular la predicción, los modos se muestran en la Tabla 2.1.

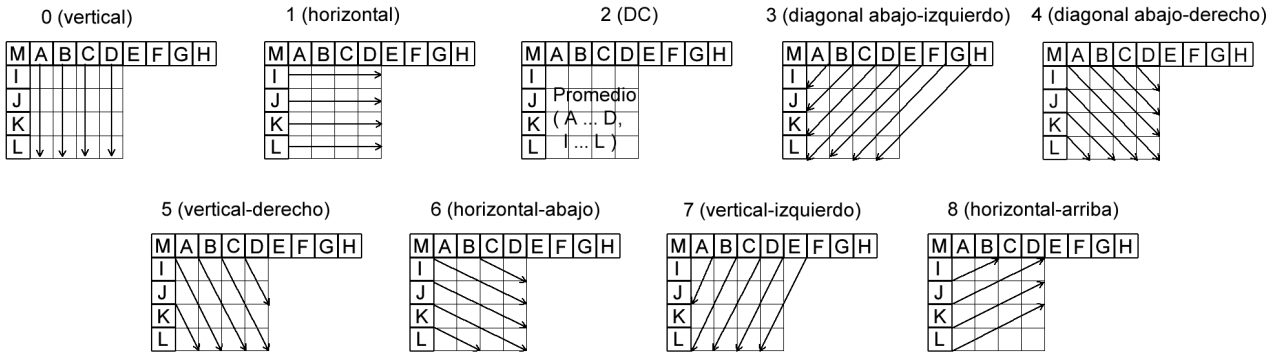


Figura 2.5: Modos de predicción para bloques de $4 \times 4$ y $8 \times 8$.

| Tamaño de bloque | Descripción |
|---|---|
| $16 \times 16$ (luma) | Para un bloque de predicción $16 \times 16$ existen cuatro modos de predicción. |
| $8 \times 8$ (luma) | Para un bloque de predicción $8 \times 8$ existen nueve modos de predicción. Sólo para perfiles High. |
| $4 \times 4$ (luma) | Para un bloque de predicción $4 \times 4$ existen nueve modos de predicción. |
| Chroma | Un bloque de predicción es generado para cada componente de croma. Existen cuatro modos de predicción. El mismo modo de predicción es usado para ambos componentes de croma. |

Tabla 2.1: Tipos de predicción de macrobloques *Intra*.

En la predicción *Intra* se usan como referencia los píxeles vecinos del macrobloque, en la Figura 2.5 se muestra los nueve modos diferentes para calcular las predicciones con bloques de tamaño $4 \times 4$, para los bloques $8 \times 8$ los modos son similares a diferencia que se hace uso de bloques de $8 \times 8$. En la Figura 2.6 se muestran los cuatro modos de predicción para bloques de $16 \times 16$.

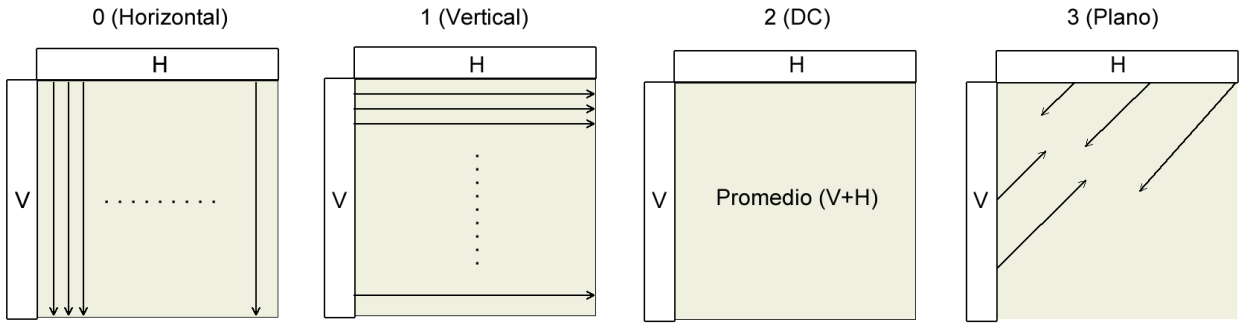


Figura 2.6: Modos de predicción para bloques de $16 \times 16$.

#### *2.1.3.2. Predicción Inter*

La predicción *Inter* es el proceso de predecir bloques a partir de píxeles pertenecientes a *frames* que han sido previamente codificados y transmitidos los cuales se almacenan en un buffer de *frames* decodificados, en la Figura 2.7 se muestra un ejemplo de predicción *Inter*. El bloque predicho puede variar desde el macrobloque completo hasta bloques de tamaño de $4 \times 4$. Además, se genera un vector de movimiento en el cual se especifica la compensación entre la posición del bloque actual y la región de predicción en el *frame* de referencia.

### 2.1.4 Transformada de coseno discreta

La DCT es una transformada ortogonal que convierte una señal digital del dominio espacial a una representación en el dominio frecuencial [43]. Además, la DCT permite expresar una imagen como una suma ponderada de funciones de coseno [41].

Sea $a \in \mathbb{R}^{\mathbb{Z}_n}$ la DCT de una dimensión de *A* se define como:

$$A(u) = \frac{2 \cdot c(u))}{n} \sum_{x=0}^{n-1} a(x) \cdot \cos\left(\frac{(2x+1)u\pi)}{2n}\right) \tag{2.1}$$

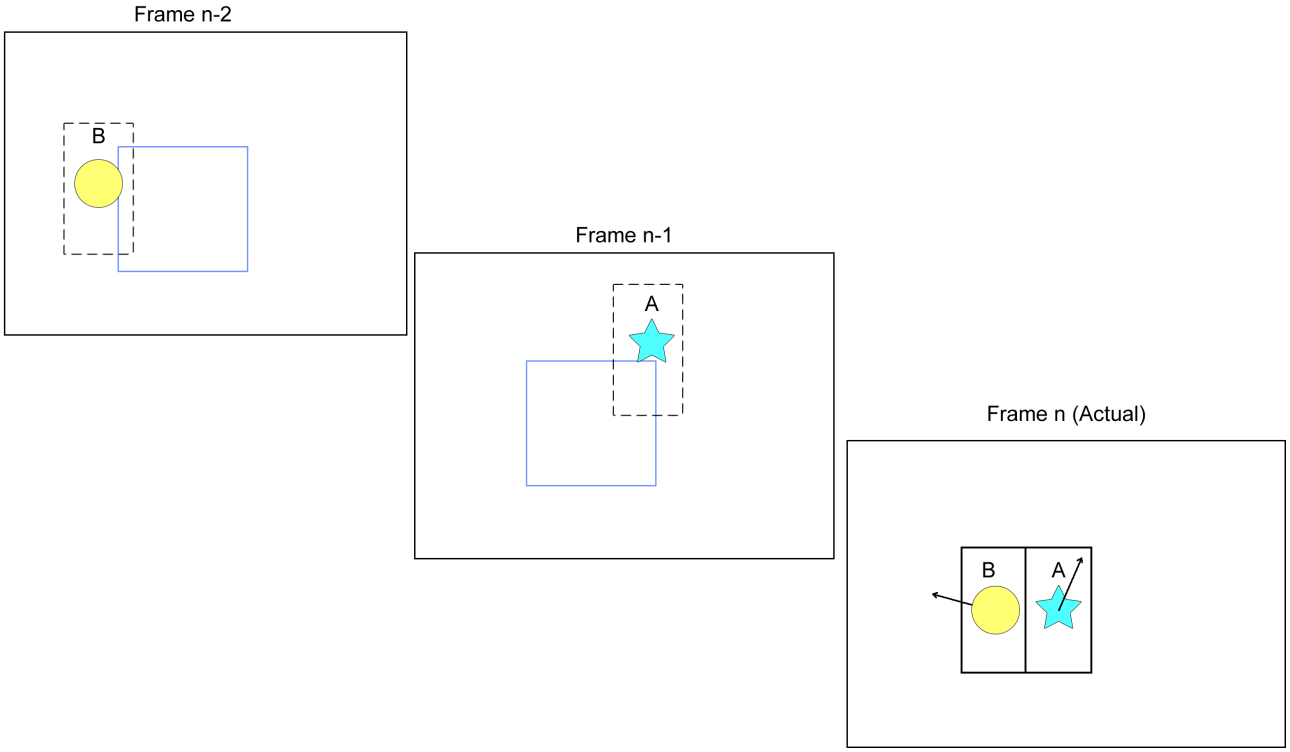


Figura 2.7: Ejemplo de una predicción *Intra*.

donde:

$$c(u) = \begin{cases} \frac{1}{\sqrt{2}} & si\ u = 0 \\ 1 & si\ u = 1, 2, \ldots n-1 \end{cases} \tag{2.2}$$

Por otra parte, la transformada de coseno discreta inversa (IDCT, por sus siglas en inglés) esta dada por:

$$A(x) = \sum_{u=0}^{n-1} c(u) \cdot a(u) \cdot \cos\left(\frac{(2x+1)u\pi}{2n}\right) \tag{2.3}$$

Una de las características más significante de la DCT es que permite compactar la energía espacial en pocos coeficientes de frecuencia. Debido a esta características la DCT es una transformada altamente usada en el ámbito de compresión de imágenes digitales. Estándares de compresión de imágenes, como JPEG [48], y video como MPEG- video [2] y MPEG-4 Visual [3] hace uso de la

transformada discreta del coseno de dos dimensiones(2-D DCT). La 2-D DCT se define como:

$$(u,v) = \frac{2 \cdot c(u) \cdot c(v)}{n} \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} a(x,y) \cdot \cos\left(\frac{(2x+1)u\pi}{2n}\right) \cdot \cos\left(\frac{(2y+1)v\pi}{2n}\right) \tag{2.4}$$

y la 2-D IDCT :

$$(x,y) = \frac{2}{n} \sum_{u=0}^{n-1} \sum_{v=0}^{n-1} c(u) \cdot c(v) \cdot a(u,v) \cdot \cos\left(\frac{(2x+1)u\pi}{2n}\right) \cdot \cos\left(\frac{(2y+1)u\pi}{2n}\right) \tag{2.5}$$

Sin embargo, el uso de la DCT suele generar algunos desajustes entre diferentes implementaciones de codificadores y decodificadores debido a la aproximación en multiplicación de factores irracionales. En atención a este problema estándares de videos anteriores especifican que la transformada inversa debe cumplir algunos criterios de exactitud basados en el estándar de la IEEE 1180-1990 [1]. En H.264/AVC el proceso de la transformada y cuantificación son diseñados para brindar eficiente codificación de video, eliminar el problema de desajuste entre el codificador y decodificador, y así, facilitar implementaciones de baja complejidad. Para lograr dicho diseño se utiliza *la transformada entera*, la cual hace uso de aritmética de números enteros de precisión limitada. En cuestión al proceso de cuantificación se integra un paso de normalización para minimizar el número de multiplicaciones requeridas para procesar un bloque de información.

#### *2.1.4.1. Proceso de transformación y cuantificación en H.264/AVC*

El proceso de transformada se realiza con una base denominada "*core*" la cual es una transformada entera de $4\times4$ o $8\times8$, y en ciertos casos, parte del resultado de la transformada entera es nuevamente transformada con una *transformada DC* utilizando una transformada Hadamard.

El proceso de trasformada por defecto para los componentes de luminancia se realiza como se muestra en la Figura 2.8, sin embargo existe dos excepciones, cuando el macrobloque es codificado utilizando una predicción Intra de $16 \times 16$ o cuando es seleccionada una transformada entera de $8 \times 8$. En el proceso por defecto cada macrobloque es dividido en 4 bloques de $4 \times 4$, cada

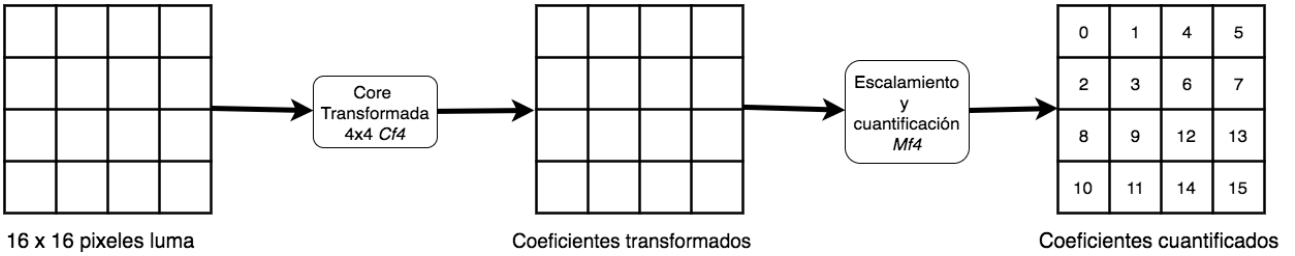


Figura 2.8: Transformada por defecto para los componentes de luminancia.

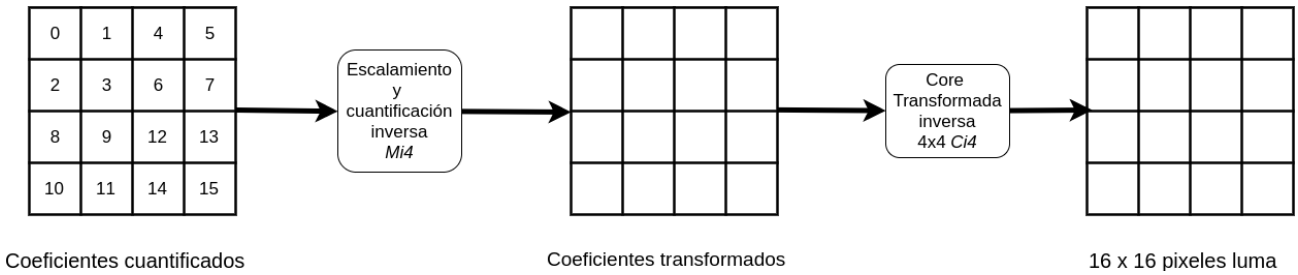


Figura 2.9: Transformada inversa por defecto para los componentes de luminancia.

uno es transformado ($C_{f4}$), escalado y cuantificado ($M_{f4}$) para producir bloques de coeficientes transformados y cuantificados de $4 \times 4$. El correspondiente proceso inverso se muestra en la Figura 2.9.

Cuando el macrobloque es predicho con bloques de predicción Intra de $16 \times 16$ una segunda transformada es aplicada para las frecuencias DC de la primera transformada. Estos valores de frecuencia suelen estar altamente correlacionados y una segunda trasformada mejora el desempeño de codificación. Después que se aplique la transformada $C_{f4}$, con los coeficientes DC de cada bloque $4 \times 4$ se forma un bloque $4 \times 4$ de coeficientes DC a los cuales se transforman usando un transformada Hadamard. El bloque de los coeficientes DC transformados y el los 15 elementos restantes de cada bloque $4$ transformado son escalados y cuantificados como se muestra en la Figura 2.10. El proceso inverso se observa en la Figura 2.11.

Para los macrobloques que son codificados con predicción Intra $8 \times 8$, que sólo está disponible en perfiles *High*, la transformada se realiza utilizando un *core* $8 \times 8$ ($C_{f8}$) y se escala y cuantifica ($M_{f8}$) como se muestra en la Figura 2.12. De manera similar el proceso inverso se muestra en la

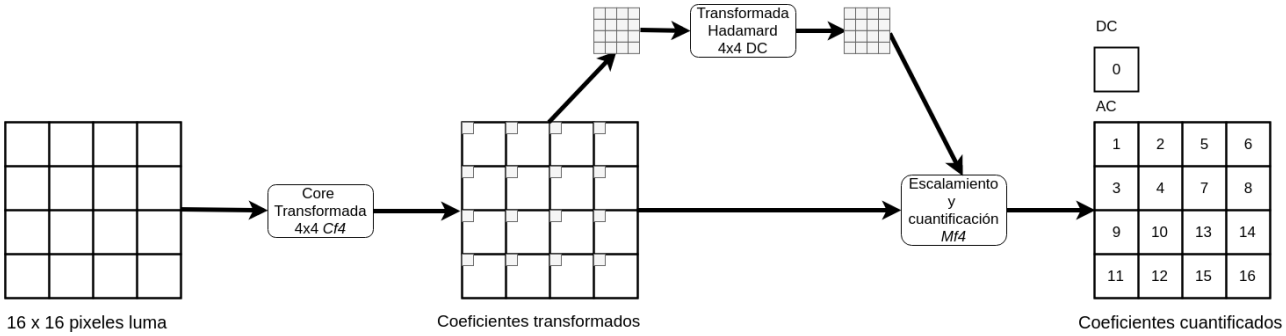


Figura 2.10: Transformada para bloques Intra de $16 \times 16$.

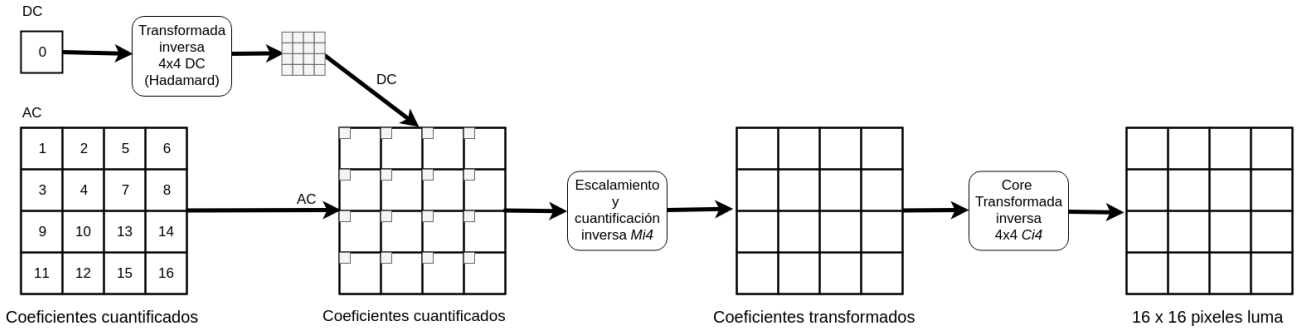


Figura 2.11: Transformada inversa para bloques Intra $16 \times 16$.

Figura 2.13.

### 2.1.5 Filtro de deblocking

Cuando un *frame* es procesado en macrobloques se tiene como resultado *artefactos de bloques* los cuales son rasgos visibles de edición en el *frame*. Para hacer frente a este problema el estándar H.264/AVC hace uso de un *filtro de deblocking* [32]. El filtro suaviza los bordes de los bloques

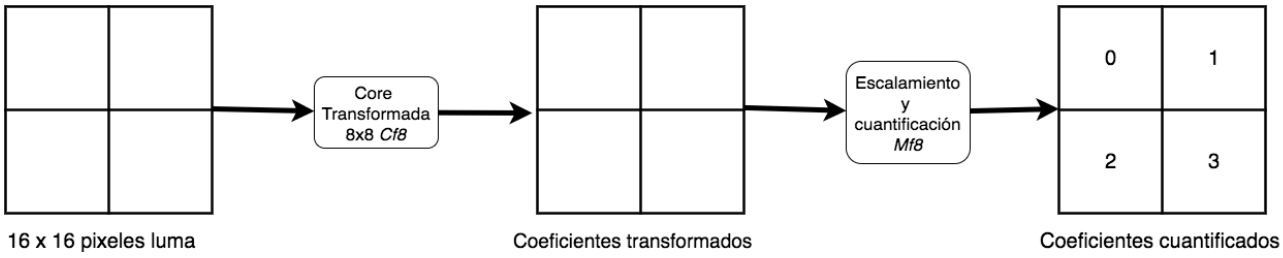


Figura 2.12: Transformada para bloques $8 \times 8$.

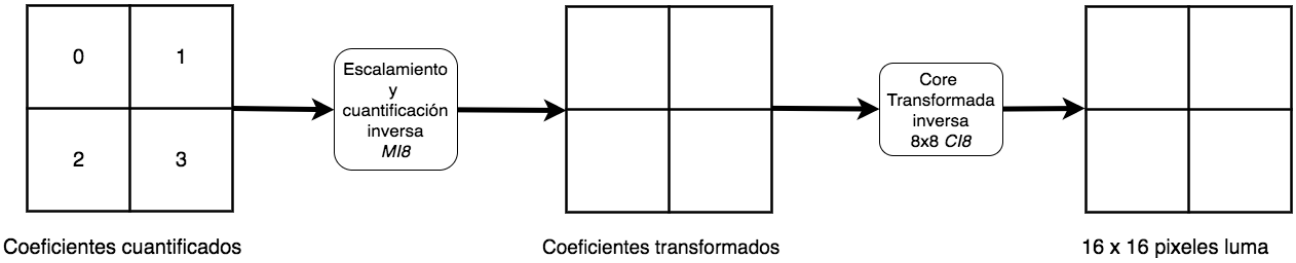


Figura 2.13: Transformada inversa para bloques $8 \times 8$.

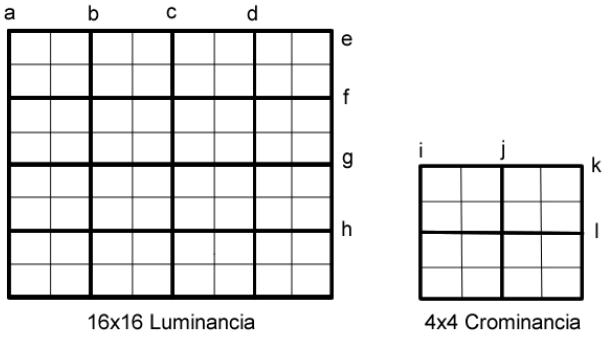


Figura 2.14: Orden de filtrado en un macrobloque.

mejorando la apariencia de los *frames* decodificados. El filtro se aplica para los componentes de luminancia y crominancia por separado. El filtro es aplicado en los bordes verticales u horizontales de los bloques $4 \times 4$ en un macrobloque en el siguiente orden.

El orden en el que se aplica el filtrado se muestra en la Figura 2.14, inicialmente los cuatro bordes verticales de los componentes de luminancia a, b, c y d son filtrados y posteriormente, los bordes horizontales de los componentes de luminancia e, f, g y h son filtrados. Finalmente los 2 bordes verticales de crominancia i, j y los bordes horizontales de crominancia k y l son filtrados [37].

La operación de filtrado afecta 3 píxeles en cada lado del límite. Las cuatro píxeles en el borde vertical o en el borde horizontal en los bloques adyacentes son p0, p1, p2, p3 y q0, q1, q2, q3, respectivamente, como se muestran en la Figura. 2.15.

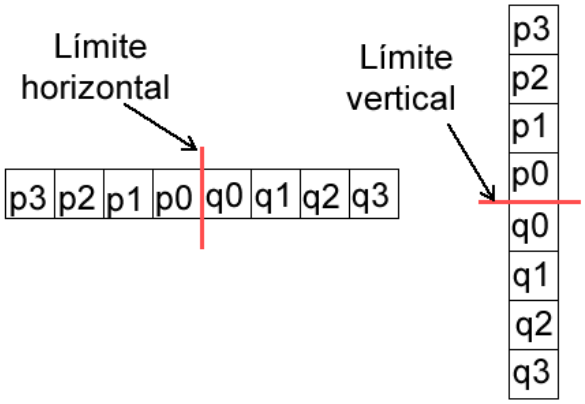


Figura 2.15: Píxeles adyacentes para bordes horizontales y verticales.

### 2.1.6 Perfiles y niveles

Dentro del estándar H.264/AVC los *Perfiles* y *Niveles* son especificaciones que proveen interoperabilidad entre implementaciones del codificador y el decodificador dentro de aplicaciones del estándar y entre varias aplicaciones que tienen requerimientos similares [35]. Específicamente el Perfil define un conjunto de características de sintaxis que se usan al generar el flujo de bits (*bitstream*), en la Figura 2.16 se muestran los seis perfiles actuales con sus características principales. Por otro lado, el Nivel coloca restricciones en ciertos parámetros del flujo de datos como el valor máximo de la tasa de bit (*bit rate*) y el máximo tamaño de imagen. En conjunto el Perfil y el Nivel limitan los requerimientos computacionales y de memoria máximos que serán usados en el decodificador.

Cada uno de los perfiles fue diseñado para ser utilizados en aplicaciones específicas [12, 23], algunas de las aplicaciones de cada perfil son:

- **Baseline:** Usado principalmente en videoconferencias y aplicaciones móviles.
- **Extend:** Usado en transmisión de vídeo y tiene una capacidad adicionales para robustez a pérdidas de datos.
- **Main:** Usado en la transmisión de televisión y algunos dispositivos portátiles.

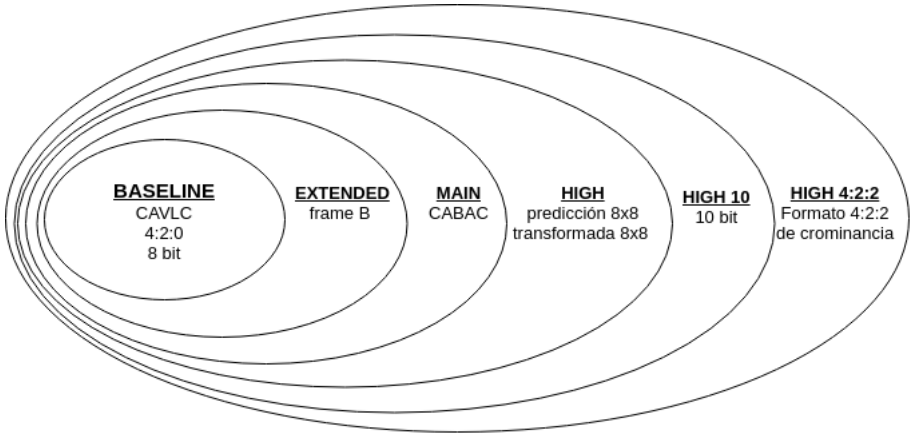


Figura 2.16: Perfiles actuales del estándar H.264/AVC.

- **High:** Agrega herramientas al perfil *Main* las cuales mejoran la eficiencia de compresión. Es usado en aplicaciones de difusión y almacenamiento de discos, en particular para aplicaciones de televisión de alta definición.

- **High 10:** Se basa en el soporte de perfil *High* para añadir hasta 10 bits por muestra de precisión de imagen decodificada.

- **High 4:2:2** Agrega soporte al perfil *High* para videos con formato 4:2:2.

En este trabajo se utiliza el perfil *Baseline*, debido a que es el utilizado en los trabajos [42, 46] y se realiza una comparación directa, y el perfil *High* debido a que es el utilizado en compresión de películas de alta definición.

### 2.1.7 Grupo de imágenes (GOP)

Como se ha mencionado, en el estándar H.264/AVC existen tres tipos de *frames*, *frame I*, *frame P* y *frame B*. El orden en el cual se ordenan se define como la estructura Grupo de Imágenes (GOP, por sus siglas en inglés). Una estructura GOP siempre inicia con un *frame I*, los *frames P* se presentan en intervalos y los *frames B* se presentan entre *frames I* y *frames P*. Los *frames I* y *frames P* son usados como referencia mientras que los *frames B* nunca son usados como referencia.

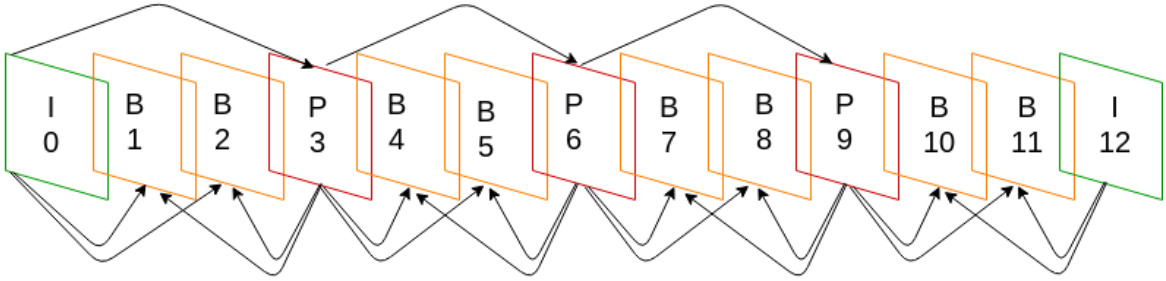


Figura 2.17: Estructura clásica de GOP.

La estructura GOP comúnmente utilizada en la compresión de videos se muestra en la Figura 2.17. En Esta estructura GOP está compuesta por 12 *frames*, iniciando con un *frame I* seguido de un *frame B*, luego un *frame P*, nuevamente dos *frames B* seguido de un *frame P*, y así sucesivamente hasta complementar 12 *frames*. Una vez completo el GOP se inicia nuevamente con el *frame I* y se sigue el mismo orden hasta complementar todos los *frames* a codificar.

## 2.2 Marcado de agua (*watermarking*)

El *marcado de agua* es un método de inserción de información dentro de un archivo multimedia como las fotografías, videos digitales y música digital. La *información* insertada o *marca de agua* puede ser identificadores del propietario, mensajes de derechos de autor, información acerca de los creadores del trabajo, etc. [5]. En el presente trabajo de tesis las secciones de marcado de agua son enfocadas a *marcado de agua de video*. En la Figura 2.18 se muestra un diagrama general de un sistema de marcado de agua digital.

### 2.2.1 Terminologías del marcado de agua

Algunas terminologías importantes pertenecientes al marcado de agua se describen a continuación:

- **Carga útil o capacidad (Payload)**.- Se refiere al número de *bits* que una marca de agua


Figura 2.18: Diagrama general de un sistema de marcado de agua digital.

codifica dentro de una unidad de tiempo o dentro de alguna otra unidad. En el caso del video digital el *payload* puede referirse al número de bits por *frame* o el número de bits por segundo.

- **Perceptibilidad**.- Es la visibilidad que puede tener una marca de agua. Las marcas de agua perceptibles son visibles al ojo humano mientras que las marcas no perceptibles no lo son.

- **Robustez**.- Es la capacidad de la marca de agua de sobrevivir a ataques tanto intencionados (maliciosos) como no intencionados (no maliciosos).

### 2.2.2 Aplicaciones del marcado de agua

Existe una gran variedad de aplicaciones del marcado de agua las cuales se aplican hoy en día, algunas de las más importantes se describen a continuación:

- **Protección de derechos de autor (Copyright)**.- Es una de las aplicaciones más utilizada. La marca de agua contiene información del propietario del archivo multimedia la cual se puede utilizar para identificar y proteger la propiedad de derechos de autor.

- **Autenticación de contenido**.- La marca de agua insertada ayuda a determinar si el archivo multimedia ha sido modificado. Si la marca de agua insertada no puede ser recuperada hay indicio de que el contenido en el archivo multimedia no es auténtico.

- ***Fingerprinting* digital**.- Un identificador único es insertado en cada archivo multimedia. Cada identificador representa a un usuario que ha adquirido un archivo multimedia, a su vez el identificador permite rastrear a los usuarios que incumplen las políticas de uso como la distribución ilegal. Esta aplicación es explicada con más detalle en la Sección 2.3 de este capítulo.

- **Monitoreo de brodcast**.- Es utilizada para publicidad y en la industria del entretenimiento para el monitoreo de contenido.

- **Aplicaciones médicas**.- Esta aplicación es usada en imágenes médicas donde se provee autenticación y confidencialidad sin afectar dicha imagen médica.

- **Control de copia**.- La marca de agua aplicada delimita el número de copias posibles que se puedan hacer al archivo multimedia en cuestión. Para realizar las copias es necesario un hardware especializado el cual modifica la marca de agua reduciendo el número de copias posibles hasta llegar al límite.

### 2.2.3 Ataques de marcado de agua

Existen diferentes ataques a los que puede estar expuesto una marca de agua, se puede distinguir en dos grupos: *intencionales* y *no intencionales*. Los ataques no intencionales son los que sufre el archivo multimedia como la compresión y ruido que pueda generar en el medio de transmisión. Los ataques intencionales son aquellos que son generados por personas con la intención de eliminar la información insertada en el archivo multimedia. Existen diferentes ataques intencionados con los que la información es parcial o totalmente eliminada, los cuales se pueden clasificar en:

1. **Ataques activos**.- En este ataque el atacante busca apoderarse de la marca de agua o simplemente hacerla indetectable. Este es un problema que afecta a aplicaciones de protección de derechos de autor y *fingerprinting*.

2. **Ataques pasivos**.- Para este tipo de ataques el atacante no busca eliminar la marca sino determinar si existe o no la marca de agua. Este tipo de ataques es una amenaza para las comunicaciones encubiertas.

3. **Ataques de colusión**.- Este ataque tiene el mismo objetivo que un ataque activo pero con un método diferente. Un conjunto de atacantes reúnen una serie de copias de la misma información. Estas técnicas son altamente usadas en contra de aplicaciones de *fingerprinting*. Este problema es tomado en cuenta en este trabajo de tesis por lo que se discute en la Sección 2.5.

4. **Ataques de falsificación**.- El atacante busca insertar una nueva marca de agua valida removiendo la existente. De esta forma se puede hacer pasar como original a un archivo pirata.

### 2.2.4 Técnicas de marcado de agua

Existen diferentes técnicas de *marcado de agua* las cuales se pueden clasificar según el dominio en el que se implementa:

**Dominio espacial.** [26]

- En este dominio la inserción se realiza modificando directamente el valor de los píxeles de cada *frame*, comúnmente los píxeles a modificar se seleccionan aleatoriamente.
- La complejidad computacional suele ser baja y de fácil implementación.
- Sin embargo, representan poca robustez ante ataques como compresión y colusión.

**Dominio frecuencial.** [5]

- Su implementación implica una complejidad computacional alta.
- Representan alta robustez ante ataques como compresión y colusión.

- Las principales transformadas utilizadas para cambiar del dominio espacial al dominio frecuencial son:

  - Descomposición de valor singular (SVD).- Es una técnica numérica que es usada en el análisis numérico para matrices diagonalizables. En esta técnica una matriz puede ser dividida en una multiplicación de tres matrices la cual es una técnica de álgebra lineal que divide a una matriz en tres vectores singulares.

  - Transformada discreta de Fourier (DFT).- Comúnmente usada en técnicas de marcado de agua para audio. En esta técnica la marca es insertada en las magnitudes de sus coeficientes ya que a diferencia de otras transformadas nos proporciona información tanto de frecuencia como de fase.

  - Transformada discreta del Coseno (DCT).- Comúnmente usado en técnicas de marcado de agua para imágenes. La DCT permite dividir en diferentes bandas de frecuencia haciendo fácil la inserción de información en las frecuencias medias donde la información no es muy visible y además evita ataques de ruido y compresión.

  - Transformada discreta Wavelet (DWT). Comúnmente usado en técnicas de marcado de agua para imágenes. Las cuales son divididas en 4 canales de frecuencia LL, HL, LH y HH, de esta forma se puede insertar la información en las medias frecuencias.

### 2.2.5 Modelos de marcado de agua

Existen diferentes formas en las cuales se pueden modelar los procesos de marcado de agua. Estos se pueden clasificar en dos grupos: *modelos basados en comunicación* y *modelos basados en geometría* [22].

#### *2.2.5.1. Modelos basados en geometría*

En los *modelos basados en geometría* los archivos multimedia, la marca de agua y los archivos marcados se pueden ver como vectores de alta dimensionalidad los cuales son llamados espacio de medios (*media space*). Alternativamente cuando se analizan algoritmos más complicados se pueden observar como proyecciones o distorsiones al *espacio de medios*. Cada espacio se ve como un *espacio de marcado*. Así el sistema puede ser visto como en termino de varias regiones y distribución probabilística en los *espacios de medios* o en los *espacios de marcado* [9].

#### *2.2.5.2. Modelos basados en comunicación*

Los *modelos basados en comunicación* describen el proceso de marcado de agua de una forma muy similar a un sistema de comunicaciones. De hecho, en esencia el *marcado de agua* es un sistema de comunicación, ya que se desea comunicar un mensaje, que es la marca de agua insertada, que transita sobre un canal el cual es el medio marcado (imagen, video o audio).

### 2.2.6 Espectro disperso

Los *modelos basados en comunicación* son los más utilizado en el estado del arte [27, 31, 33], específicamente con la técnica de *espectro disperso*. En comunicaciones, el espectro disperso se utiliza para transmitir diferentes señales de banda estrecha sobre una señal de ancho de banda mucho mayor, de tal forma que se presenta como una sola señal [10]. De igual forma se puede utilizar como técnica de marcado de agua donde el archivo multimedia representa la señal anfitrión y la marca de agua se extiende sobre todas las frecuencia de modo que la energía es muy pequeña y ciertamente indetectable [45]. El espectro disperso tiene dos características importantes para el marcado de agua. Primero, la energía de la señal a insertar en cualquier frecuencia es muy pequeña lo que reduce el riesgo de artefactos perceptibles. Segundo, el hecho de que la marca sea dispersa a lo largo de un gran número de frecuencias permite robustez hacia muchas distorsiones comunes que sufren las señales

[9]. La técnica de espectro disperso es altamente resistente a los ataques de colusión cuando las marcas de agua tienen una distribución gaussiana y son estadísticamente independientes [17]. Por esta razón comúnmente se usan generadores pseudoaleatorios con distribución uniforme, siendo la señal pseudoaleatoria generada la "llave" para extraer la marca de agua. Otro aspecto importante a considerar en la técnica de espectro disperso son las frecuencias a utilizar para realizar la inserción de la marca, ya que en la práctica no se suele usar todas las frecuencias, esto debido a que un simple filtro pasa bajas puede eliminar información de la marca de agua. En [10] se propone que los coeficientes que son perceptualmente significantes son viables para la inserción, por el contrario aquellos que son perceptualmente insignificantes no son viables.

## 2.3 Fingerprinting digital

Como se ha mencionado anteriormente, el *fingerprinting* es una aplicación del marcado de agua en la cual la marcada de agua es representado por un identificador (ID) de usuario. Si por algún motivo un archivo multimedia se encuentra distribuyendo de manera ilegal (copia pirata) basta con extraer el identificador para determinar que usuario adquirió dicho archivo multimedia y por lo tanto es culpable de dicho acto ilícito. En la Figura 2.19 se muestra el proceso de inserción del *fingerprinting*. Tal como se ilustra en la Figura 2.20, en una aplicación de *fingerprinting* existen tres características del marcado de agua que se explican a continuación:

- **Capacidad**.- Es el número de bits del identificador que son codificados dentro de una unidad de tiempo o espacio. Esta característica se puede ver como el número de usuarios que son posible registrar.

- **Robustez**.- La habilidad de detectar el identificador aún después de sufrir un ataque. La robustez se puede medir con el número máximo de usuarios detectados en un video pirata.

- **Transparencia**.- Es la similitud perceptual entre el archivo multimedia original y el archivo

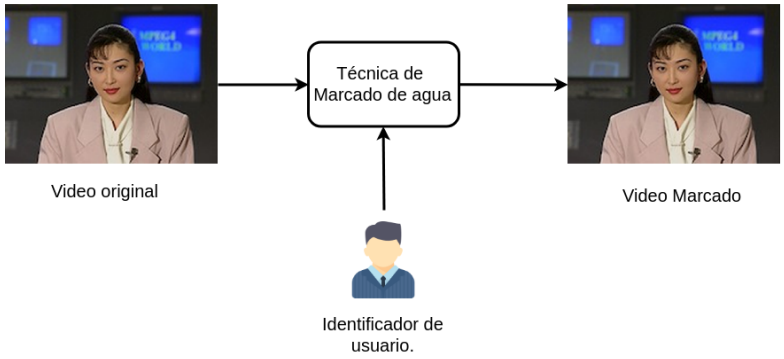


Figura 2.19: Proceso de inserción en el *fingerprinting*.

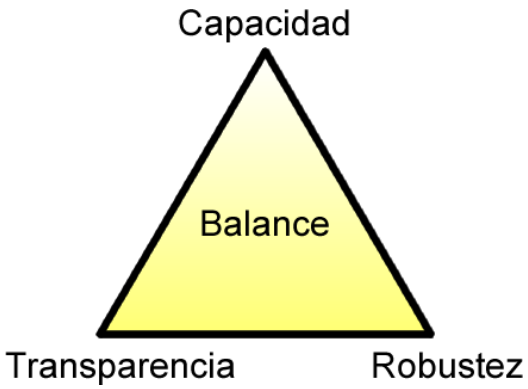


Figura 2.20: Balance entre la transparencia, robustez y capacidad.

multimedia que tiene el identificador, es decir, la calidad visual (para el caso de imágenes y video) o la calidad auditiva (en audio) que presente el archivo multimedia marcado.

Sin embargo, estas tres características están relacionadas entre sí ya que si se desea tener mayor transparencia, la capacidad será menor y/o no se tendrá mucha robustez. En cambio, si se pone énfasis en la robustez la transparencia disminuye y/o la capacidad será mínima. Por otro lado, mientras la capacidad aumenta suele verse afectada tanto la robustez como la transparencia. Debido a esto es necesario tener en cuenta un balance entre estas tres propiedades al definir la energía y longitud del identificador.

## 2.4 Métricas para evaluar la calidad visual de imágenes

Cuando una señal (para este trabajo un *frame* de video) es editada comúnmente se busca un métrica que permita medir cuantitativamente la calidad de la señal editada con respecto a la original. A continuación se describen dos métricas de calidad visual que son usadas en este trabajo de tesis.

### 2.4.1 Relación señal a ruido de pico

Una de las métricas más utilizadas en el ámbito de las señales es la *Relación Señal a Ruido de Pico* (PSNR, por sus siglas en inglés) [21]. Dicha métrica hace uso de la formulación del error cuadrático medio (MSE, por sus siglas en inglés) utilizando como unidades de medida los decibelios (dB)[20]. Dada una imagen de referencia *f* y una imagen de una prueba *g*, ambas de tamaño $M \times N$, el valor PSNR entre *f* y *g* está dado por:

$$PSNR(f,g) = 10\log_{10}(255^2/MSE(f,g)) \tag{2.6}$$

dónde:

$$MSE(f,g) = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(f_{ij} - g_{ij})^2 \tag{2.7}$$

### 2.4.2 Métrica de percepción sin referencia (QNR)

Por otra parte en [53] se propone una métrica la cual, además de no necesitar la imagen original como referencia, su valor cuantitativo es calculado con base a la percepción del ojo humano. Dicha métrica es modelada a partir de pruebas con personas las cuales calificaron una serie de imágenes que fueron comprimidas a diferentes tasas de compresión. La métrica se basa en dos de las características identificables que sufre una imagen con la compresión, el efecto de distorsión (*blurring*) y los artefactos de bloque (*blocking artifacts*). El efecto de distorsión se da debido a la

pérdida de coeficientes DCT de frecuencias altas, y los artefactos de bloque se presentan debido a la discontinuidad entre los bordes de los bloques, ya que compresores como JPEG y H.264/AVC son basados en bloques los cuales son cuantificados independientemente. Las características son calculadas tanto horizontalmente como verticalmente. Sea $x(m,n)$ una señal de una imagen para $m \in [1,M]$ y $n \in [1,N]$ y una señal de diferencia a lo largo de cada línea horizontal:

$$d_h(m,n) = x(m,n+1) - x(m,n), n \in [1,N-1] \tag{2.8}$$

Primero se determina el efecto de bloque es estimado como las diferencias del promedio entre los límites de los bloques:

$$B_h = \frac{1}{M([N/8]-1)} \sum_{i=1}^{M} \sum_{j=i}^{[N/8]-1} |d_h(i,8j)| \tag{2.9}$$

Segundo, el efecto de distorsión se puede calcular de manera fácil en el dominio de la frecuencia haciendo uso de la transformada rápida de Fourier (FFT, por sus siglas en Inglés) [52], pero debido a que la FFT se calcula demasiadas veces resulta ser computacionalmente costoso. El efecto distorsión causa una reducción en la actividad de la señal y ésta se puede medir usando dos factores. Primero se calcula la diferencia absoluta media entre las muestras de los bloques de imagen:

$$A_h = \frac{1}{7}\left[\frac{8}{M(N-1)} \sum_{i=1}^{M} \sum_{j=1}^{N-1} |d_h(i,j)| - B_h\right] \tag{2.10}$$

El segundo factor es la tasa de cruce-cero (ZC, por sus siglas en Inglés). Se define para $n \in [1, N-2]$

$$z_h(m,n) = \begin{cases} 1 \;\; horizontal\ ZC\ en\ d_h(m,n) \\ 0 \;\; de\ otra\ manera \end{cases} \tag{2.11}$$

Por lo tanto, la tasa ZC horizontal puede ser estima como:

$$Z_h = \frac{1}{M(N-2)} \sum_{i=1}^{M} \sum_{j=1}^{N-2} zh(m,n) \tag{2.12}$$

Usando los métodos similares se calcula las características verticalmente de $B_v$, $A_v$, y $Z_v$, finalmente con todas las características se tiene:

$$B = \frac{B_h + B_v}{2}, A = \frac{A_h + A_v}{2}, Z = \frac{Z_h + Z_v}{2} \tag{2.13}$$

Una forma para encontrar un buen desempeño de predicción es dado por:

$$S = \alpha + \beta B^{\gamma 1} A^{\gamma 2} Z^{\gamma 3} \tag{2.14}$$

donde: $\alpha$, $\beta$, $\gamma 1$, $\gamma 2$, y $\gamma 3$ son parámetros que son estimados con datos de prueba subjetivos utilizando una técnica de regresión lineal. El valor de los parámetros encontrados con todas las imágenes de pruebas son:$\alpha = -245.9$, $\beta = 261.9$, $\gamma 1 = -0.0240$, $\gamma 2 = 0.0160$, y $\gamma 3 = 0.0064$.

## 2.5 Ataques de colusión

Como se ha dicho anteriormente, uno de los principales ataques que afecta al *fingerprinting* son los ataques de colusión. Existen diferentes ataques de colusión los cuales se dividen en dos tipos: lineales y no lineales. En los ataques lineales destaca el *ataque promedio* por su simpleza y efectividad. Mientras que los ataques no lineales hacen uso de funciones como mínimo, máximo y mediana para generar los ataques [46].

Sea $K$ el número de usuarios deshonestos, $S_c$ el conjunto que contiene las copias de los atacantes y $X_i$ la copia del *i-ésimo* usuario. Los tipos de ataques se definen como:

$$\textbf{Promedio:} V^{ave} = \sum_{i \in S_C} X_i / K \tag{2.15}$$

$$\textbf{Mínimo:} V^{min} = \min_{i \in S_C} \{X_i\} \tag{2.16}$$

$$\textbf{Máximo:} V^{max} = \max_{i \in S_C} \{X_i\} \tag{2.17}$$

$$\textbf{Mediana:} V^{med} = \text{median}_{i \in S_C} \{X_i\} \tag{2.18}$$

$$\textbf{MinMax:} V^{minmax} = (V^{min} + V^{max})/2 \tag{2.19}$$

$$\textbf{Negativo modificado:} V^{modneg} = V^{min} + V^{max} - V^{med} \tag{2.20}$$

## 2.6 Códigos anticolusión de *fingerprinting*

Para hacer frente a los ataques de colusión se han diseñados códigos anticolusión (ACC, por sus siglas en Inglés) capaces de soportar este tipo de ataques, de tal forma que es posible determinar en su totalidad o parcialmente el número y la identidad de cada uno de los usuarios que participaron en el ataque.

Uno de los primeros trabajos en el diseño de códigos de *fingerprinting* binarios resistentes a colusión fue presentado por Boneh y Shaw en 1955 [6], en el cual se considera que los códigos satisfacen un principio subyacente denominado la *suposición de marcado*. El identificador del *fingerprinting* consiste en una colección de marcas en la que cada una es modelada como una posición en un objeto digital y puede tomar un número finito de estados. Una marca se considera detectable cuando un conjunto de usuarios no tiene la misma marca en esa posición (como se muestra en la Figura 2.21). Bajo este criterio se usó un diseño jerárquico y técnicas aleatorias para construir *c-seguros códigos* que son capaces de detectar hasta *c* usuarios coludidos entre sí con gran probabilidad. Donde la longitud reportada para una colusión de *c* usuarios está dada por $m = (c^4 \log(1/\varepsilon) \log(n/\varepsilon))$ para *n* usuarios.

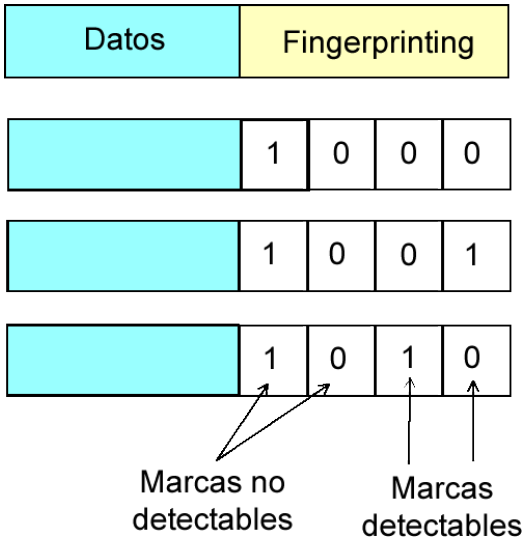


Figura 2.21: Ilustración de suposición de marcado.

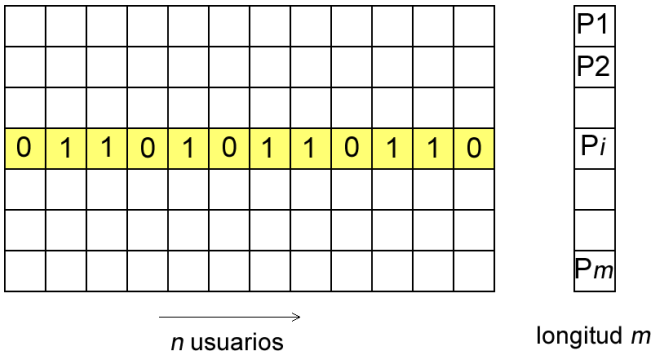


Figura 2.22: Matriz de códigos de Tardos.

Trabajos más recientes en el diseño de códigos *fingerprinting* [50], presentan los códigos de Tardos diseñados por Gábor Tardos en el 2008, los cuales son códigos binarios representados en una matriz (Figura 2.22) de $n \times m$ donde $n$ representa el número de usuarios y $m$ la longitud del código. Cada código se genera con una probabilidad aleatoria e independiente $\{p(i)\}_{1 \leq i \leq m}$, con una distribución $P$ [39]. La longitud de los códigos se definen como: $M = 100c^2 \ln(1/\varepsilon_1)$. Donde $c$ = el números de máximo de infractores y $\varepsilon_1$ = la probabilidad de falso positivo.

## 2.7 Resumen

En este capítulo se presentó una descripción detallada del estándar de compresión H.264/AVC, algunas de sus características, así como las ventajas que presenta con respecto a compresores anteriores. De igual forma, se definió la DCT y se discute el porqué de su importancia en la compresión de imágenes y video. Así mismo, se presentó la transformada entera que es una aproximación a la DCT pero calculada con un costo computacional bajo. Se explicó las diferentes formas en que se aplica la transformada entera en el estándar H.264/AVC. Además, se presentó una descripción general de la tecnología del marcado de agua digitales. Se describió algunas de las aplicaciones del marcado de agua así como los principales ataques que afectan a éstas. De manera puntual se describió la aplicación de *fingerprinting* presentando las propiedades que la componen, dos métricas de evaluación visual y algunos códigos representativos anticolusión que se han desarrollado.

# 3

# Estado del arte

En este capítulo se describe las métodos *fingerprinting* encontradas en la literatura aplicadas en diferentes archivos multimedia, aplicaciones exclusivas para video (video *fingerprinting*) y aplicaciones de video *fingerprinting* efectuadas exclusivamente al estándar H.264/AVC.

## 3.1 Fingerprinting en archivos multimedia

Uno de los trabajos pioneros en *fingerprinting* se presenta en [7], en el que se discuten algunos métodos para la asignación de palabras código (*codeword*) con el propósito de rastrear información digital como lo es el audio, video, software, documentos, etc.

Algunos trabajos generalizan la técnica de *fingerprinting* para aplicarla a cualquier tipo de archivo multimedia. En [54] se propone una técnica de *fingerprinting* de distribución Gaussiana. Dos de las características que se consideran importantes al analizar un archivo atacado son, además del máximo número de usuarios traidores, la probabilidad de *falsos positivo* y *falsos negativos*. Entendiendo como falso positivo la probabilidad de acusar a usuarios que no son culpables y falso negativo aquellos

usuarios que no se pudieron detectar y son culpables. En ambos casos la probabilidad debe ser la menor posible y se considera más crítica la probabilidad del falso positivo. En este trabajo se hace uso de un método de marcado de agua de espectro disperso y se consideran marcas Gaussianas independientes lo cual ayuda a disminuir la probabilidad de falso positivo.

En [28] se presenta un método *fingerprinting* para imágenes en el cual se utiliza una método de marca de agua basado en el espectro disperso con una técnica de acceso múltiple por división de código (CDMA, por sus siglas en Inglés), utilizando un identificador representado por una señal creado a partir de la transformada inversa DCT de un vector de una dimensión. Así mismo, se implementa una técnica de agrupación de usuarios en la cual el identificador se divide en dos partes, identificador de grupo e identificador de usuario lo cual permite tener un mayor número de usuarios.

Dicho método es adaptado para una aplicación de fingerprinting en documentos de texto en [38] y de igual forma en una aplicación *fingerprinting* de audio en [15]. En ambos trabajos se reportan resultados competitivos. Sin embargo, en [44] se proponen dos ataques certeros en el cual se reporta que la efectividad del método *fingerprinting* disminuye considerablemente. Para hacer frente a estos nuevos tipos de ataques en [29] se presenta una contra medida que se basa en hacer uso del canal de luminancia, y antes de realizar la inserción del identificador multiplicar la imagen (en el dominio espacial) por una matriz pseudoaleatoria (con valores -1 o 1) con distribución uniforme del mismo tamaño de la imagen, una vez que se ha hecho la inserción y la imagen es trasformada al dominio espacial, nuevamente se multiplica por la misma matriz pseudoaleatoria utilizada.

## 3.2 Fingerprinting en video

Uno de los primeros enfoques de marcado de agua resistente a colusión es propuesto en [49]. Se utiliza una técnica de inserción de *frame* por *frame*. Se crea una patrón básico de marca de

agua el cual es insertado repetidamente de modo que es centrado alrededor de un número fijo de puntos seleccionados, conocido como puntos de anclaje, en cada *frame* del video. La parte del *frame* donde es insertada la marca de agua es conocida como huella (footprint). Los puntos de anclaje son calculados mediante un algoritmo de extracción de características. Debido a esto a medida que cambia el contenido del *frame* de video también lo hacen los puntos seleccionados. Como resultado de esa marca de agua las huellas evolucionan con el video. Después de generar esas marcas para los *frames* de la señal anfitrión (*frame* del video original) se aplica enmascaramiento espacial para obtener criterios de robustez e imperceptibilidad. A continuación, la marca de agua a escala se inserta en los datos de la señal anfitrión mediante una adición. Sin embargo las pruebas realizadas sólo reportan ataques de colusión lineal (promedio).

En [25] se propone un esquema que se basa en la DCT el cual es perceptualmente invisible además de ser robusto contra ataques de rotación y de colusión. Para hacer el esquema resistente a rotaciones el identificador es insertado en bloques cuadrados, colocados en medio sobre el canal de luminancia. En cada bloque seleccionado se calcula los momentos de Zernike. Para que este esquema sea robusto contra ataques de colusión, se utilizan bloques de $8 \times 8$ seleccionados aleatoriamente por lo que cada bloque seleccionado cambian para las tramas sucesivas del video. Para ello se utiliza un número pseudoaleatorio en donde la semilla es obtenida de un vector de permutaciones. Sin embargo, el identificador utilizado es un logotipo (imagen) binaria la cual es vectorizada. Para un escenario real el uso de imágenes no es tan factible debido a que se necesita un gran número de usuarios se tendría que implementar un generador de imágenes binarias la cuales sean lo menos correlacionadas entre sí. En dicho esquema la marca se inserta antes de la codificación MPEG-4 y la extracción después de la decodificación.

Otra propuesta, en [36], es usar códigos de respuesta rápida (QR, por sus siglas en Inglés) debido a sus características principales: uso de plantillas inherente, resistencia a ruido y tamaño compacto. Esta propuesta usa una técnica de marcado de agua basado en escena reportado en [8], donde la marca se realiza a lo largo de cada escena. La inserción se realiza mediante las selecciones de regiones

de energía media cuadrática (RMS, por sus siglas en Inglés) mínimas y divididas en bloques de $8 \times 8$. El uso de códigos QR permite que el número de usuarios disponibles sea muy grande ($10^{11}$). Sin embargo, si existe un ataque de inserción o eliminación de *frame* debe realizarse una sincronización de video. Además, no reporta alguna robustez a compresión con perdida.

Otra propuesta es [19], en la cual se pone énfasis en el número de usuarios ya que, como se ha mencionado, en un escenario real el número de usuarios ronda los millones. Se emplea un código *q-ary* corrección de errores (ECC, por sus siglas en Inglés) construido como un código resistente a colusiones. Además utiliza una codificación e inserción conjunta (*Joint Coding and Embedding*) presentada en [18]. Un *framework* típico de ECC incluye una capa de código una capa de inserción basado en espectro disperso. Para insertar el identificador se divide el video en segmentos, cada segmento puede ser un *frame* o un conjunto de *frames*. Antes de insertar la secuencia de espectro se aplica la técnica de inserción posterior permutada (PSE, por sus siglas en Inglés) propuesta en [18] para mejorar la resistencia a la colusión. En PSE cada segmento del identificador es particionado en diferentes subsegmentos los cuales son permutados aleatoriamente de acuerdo con una llave secreta. Los resultados reportados son con base a la probabilidad de detectar al menos uno de los usuarios coludidos entre sí, por lo que reporta probabilidad de detección con hasta 100 usuarios coludidos entre sí.

En Tabla 3.1 se muestra una comparación de trabajos que usan técnicas de *fingerprinting* en formatos diferentes a H.264/AVC. En dichos trabajos hay técnicas que se pueden retomar para generar mayor robustez así como descartar otras por presentar alguna desventaja. Por ejemplo, en [49] el uso de inserción *frame* por *frame* permite robustez a ataques de descarte de *frame*. En [25], aunque es común, utilizar imágenes como marcas, en un escenario *fingerprinting* no es factible debido a que se necesita un gran número de usuarios, se tendría que crear una imagen por cada usuario y las imágenes no deben estar correlacionadas entre sí. Por otra parte, en [36] se tiene un gran número de usuarios disponibles, ideal para un escenario real de *fingerprinting*, sin embargo un ataque de descarte de *frame* es necesario una sincronización de *frames*.

Tabla 3.1: Comparativa de trabajos que abordan la detección de copias de video pirata en formatos diferentes a H.264/AVC. NE=No especificado.

| Trabajo | Técnica de inserción | Tipo de identificador | Tipo de colusión | Número de infractores detectados* |
|---|---|---|---|---|
| Su *et al.* (2002) [49] | *frame* por *frame* con selección de puntos de anclaje. | Patrón básico | Promedio | NE |
| Karmakar *et al.* (2015) [25] | Inserción en coeficientes DCT pseudoaleatorios en bloques de $8 \times 8$ | Imagen binaria | Promedio | NE |
| Metha *et al.* (2014) [36] | Selección de región de energía mediante RMS basada por escena. | Código QR | Promedio | 20 |
| He *et al.* (2007) [19] | Espectro disperso. | Identificadores ECC | Promedio, mínimo, MinMax | Probabilidad de 100 % de detectar un usuario con hasta 100 coludidos |

*Cuando se obtiene el 100 % de detección de coludidos.

## 3.3 Fingerprinting en video comprimido con el estándar H.264/AVC

En el estado del arte existen varios trabajos que utilizan técnicas de fingerprinting y marcado de agua resistentes a ataques de colusión, sin embargo, escasos trabajos hablan de la implementación de estas técnicas en el estándar H.264/AVC.

En [34] se presenta una técnica de marca de agua *frame* por *frame*. Para lograr mayor robustez y mantener la calidad se combinan las técnicas de codificación truncada por bloque (BTC, por sus siglas en Inglés), transformada discreta wavelet (DWT, por sus siglas en Inglés) y descomposición del valor singular (SVD, por sus siglas en Inglés). La marca de agua se obtiene utilizando la técnica de BTC con la que se obtiene una imagen binaria. Por medio de la trasformada Wavelet cada *frame* del video se divide en las cuatro bandas de resolución (LL, LH, HL, LL). A la banda de alta resolución (HH) se le aplica el proceso SVD obteniendo tres matrices diferentes (*S, V, D*). Los valores de la marca son insertados en la matriz *S*. Para obtener la marca, el proceso es similar al de la inserción, en donde en lugar de insertar los valores de la matriz *S*, estos son extraídos para formar la marca. Una de las ventajas que presenta este trabajo es que cada *frame* contiene la marca de agua, hecho que debería degradar la calidad del video, sin embargo, los resultados reportados demuestran que sus valores de Relación Señal a Ruido Pico (PSNR) están por encima de otras técnicas. Otra ventaja de tener la marca en cada *frame* es que no existe un número mínimo de *frames* para detectar la marca.

Por otra parte, en [46] se propone como *fingerprinting* el uso de códigos probabilísticos de Tardos insertados en el estándar H.264/AVC usando una técnica de *watermarking* basada en el espectro disperso. Cada bit del código de Tardo se inserta en un bloque $4\times4$ *Intra* mediante el uso de la técnica de espectro disperso en los coeficientes DC y AC que tienen una magnitud por encima de un cierto umbral *TH*. El código de Tardos es modulado con una secuencia Gaussiana bipolar y escalado por

un factor $\alpha$. Para la extracción del código es necesario contar con el valor del umbral *TH*, mediante una correlación lineal se obtienen los bits, si el valor de la correlación es positiva el bit extraído tiene el valor de 1, de lo contrario es igual a 0. La ventaja más significativa de este trabajo se basa en los resultados obtenidos con los códigos de Tardos ya que en las pruebas realizadas la detección correcta se reporta entre el 90 % y 95 % de los infractores en el ataque de colusión. Sin embargo, la longitud del código de Tardos crece cuadráticamente con respecto al número de infractores a detectar.

De manera similar en [42] se usan los códigos de Tardos pero esta vez se presenta un esquema embebido adaptativo robusto usando un modelo de distorsión apenas perceptible (JND, por sus siglas en Inglés). En este enfoque el proceso de inserción es realizado durante el proceso de codificación en la selección de macrobloques de tipo Intra 4x4 dentro del *frame I*. El código *fingerprint* es insertado en los coeficientes AC mediante el uso de técnicas de espectro disperso, por otra parte el modelo JND ajusta dinámicamente la fuerza y así lograr ser impredecible. El código *fingerprint* binario es mapeado en el código bipolar (-1,1) para obtener valores polares para ser modulados por una secuencia del espectro disperso. La extracción del código de Tardos se obtiene de la decodificación H.264/AVC mediante un proceso de cuantificación inversa. Los bits primeramente son extraídos de forma bipolar mediante una corrección lineal para posteriormente ser mapeados a un código binario. Este trabajo presenta las mismas ventajas que el trabajo anterior [46] añadiendo el hecho de que usa un una técnica JDN con lo cual calidad del video se degrada menos permitiendo soportar más usuarios.

En la Tabla 3.2 se muestran las características principales de los trabajos que implementan técnicas de *fingerprinting* en el formato H.264/AVC. Por otra parte, en la Tabla 3.3 se muestran los resultados reportados para los diferentes ataques en los trabajos [46] y [42]. En ambos trabajos se tiene una detección de 100 % para hasta 8 usuarios coludidos entre sí y conforme aumenta el número de usuarios coludidos entre sí el porcentaje de detección disminuye. Para pruebas mayores a 8 usuarios coludidos entre sí los resultados de detección son similares y en algunos casos la propuesta [42] presenta un detección mayor de uno o dos usuarios.

Tabla 3.2: Comparativa de trabajos que abordan la detección de copias de video pirata en formato H.264/AVC.

| Trabajo | Técnica de inserción | Tipo de identificador | Tipo de colusión | Desventaja |
|---|---|---|---|---|
| Manaf *et al.* (2016) [34] | Transformada wavelet | Imagen | Promedio entre archivos diferentes. Promedio entre archivos del mismo usuario. | El uso de una marca diferente en cada *frame* para un escenario real aumentaría exponencialmente el número de marcas necesarias. |
| Shahid *et al.* (2013) [46] | Espectro disperso | Códigos de Tardos | Promedio, mínima, máxima, mediana, MinMax. | La longitud del código de Tardos crece cuadráticamente con respecto al número de infractores a detectar. No es robusto a ataque de descarte de *frame* debido a que el identificador está a lo largo de distintos *frames* |
| Saadi *et al.*(2014) [42] | Espectro disperso utilizando un modelo JND | Códigos de Tardos | Promedio, mínima, máxima, mediana, MinMax | La longitud del código de Tardos crece cuadráticamente con respecto al número de infractores a detectar. |

Tabla 3.3: Resultados de detección de usuarios coludidos entre sí reportados de la propuesta (1), Shahid y (2) Saadi

| Propuesta | Número de Coludidos | Número de usuarios detectados tipos de colusión | | | | |
|---|---|---|---|---|---|---|
| | | Avg | Min | Max | Med | MinMax |
| 1 | 2 | 2 | 2 | 2 | 2 | 2 |
| 2 | | 2 | 2 | 2 | 2 | 2 |
| 1 | 5 | 5 | 5 | 5 | 5 | 5 |
| 2 | | 5 | 5 | 5 | 5 | 5 |
| 1 | 8 | 8 | 8 | 8 | 8 | 8 |
| 2 | | 8 | 8 | 8 | 8 | 8 |
| 1 | 11 | 11 | 10 | 10 | 10 | 10 |
| 2 | | 10 | 10 | 10 | 10 | 10 |
| 1 | 14 | 14 | 13 | 13 | 13 | 13 |
| 2 | | 14 | 14 | 14 | 13 | 14 |
| 1 | 17 | 16 | 15 | 14 | 14 | 14 |
| 2 | | 15 | 16 | 16 | 16 | 15 |
| 1 | 20 | 18 | 16 | 16 | 16 | 17 |
| 2 | | 17 | 17 | 17 | 16 | 17 |

## 3.4 Discusión

Debido a que el estándar H.264/AVC utiliza una transformada entera y considerando que las inserciones de la marca en el dominio frecuencial presenta una mayor robustez a las inserciones en el dominio frecuencial, e considerable el uso de una técnica que utilice una transformada entera o en su defecto una DCT. El uso de una método de espectro disperso como técnica en el marcado de agua ha demostrado robustez contra diferentes ataques. La inserción del identificador *frame* por *frame* es otra técnica a considerar para generar robustez a ataques de descarte de *frame*. Teniendo en cuenta que la inserción del identificador en frecuencias bajas ayuda a la robustez contra la perdida de información se puede hacer uso de una técnica como la WDT para dividir el canal de frecuencias bajas.

## 3.5 Resumen

En este capítulo se describieron los métodos relevantes de *fingerprinting* reportados en la literatura donde se detallan las técnicas de marcado de agua que se utilizan así como el identificador (marca de agua). En la primer sección se describieron los métodos *fingerprinting* para archivos multimedia en general donde destaca el trabajo presentado en [28] ya que se ha implementado tanto en imágenes, documentos y audio reportando buen desempeño. En una segunda sección se presentaron los métodos fingerprinting aplicados a videos codificados en un estándar diferente al H.264/AVC, en donde se encuentra que el uso de inserción de *frame* por *frame* ofrece robustez adicional. En un tercera sección se describieron pocos trabajos que proponen un método *fingerprinting* implementado a videos en formato H.264/AVC resistente a colusión.

# 4

# Método propuesto

En este capítulo se detalla el método propuesto describiendo la técnica de marcado de agua y el identificador seleccionado así como la adaptación al estándar H.264/AVC.

## 4.1 Diseño e implementación de la propuesta

En [28] es propuesto un identificador basado en una secuencia de espectro disperso utilizando secuencias cuasi-ortogonales moduladas por un ruido pseudoaleatorio. El uso de una secuencia cuasi-ortogonal, obtenidas mediante una transformada ortogonal como la DCT, permite generar fácilmente identificadores que no estén correlacionados entre sí, problema que se presenta en algunas técnicas. Además, el uso de señales de ruido pseudoaleatorias permite utilizar un enfoque de agrupamiento para obtener códigos identificadores de longitud reducida. Por dichas razones se decide usar este tipo de identificador. Teniendo en cuenta que el estándar H.264/AVC en el proceso compresión/descompresión utiliza la transformada entera, se propone utilizar dicha transformada para la inserción de los identificadores, utilizando una técnica de marcado de agua de espectro disperso

para aprovechar la robustez que ofrece dicha técnica.

Dado que el identificador utiliza un enfoque de agrupamiento, es decir, el identificador es formado por un ID de grupo y por un ID de usuario, de esta forma primero se genera el ID de grupo y posteriormente el ID de usuario. El identificador será insertado *frame* por *frame* para generar mayor robustez a ataques como inserción y descarte de *frames*. El proceso de generación, inserción y detección se describen a continuación. Por otro lado, el proceso completo del método propuesto en este trabajo se presenta en la Figura 4.1.

### 4.1.1 Generación del identificador

Como se mencionó anteriormente, el identificador se comprende de dos partes, el identificador del usuario e identificador de grupo, ambos identificadores se crean por separado de manera similar.

1. El identificador es creado a partir de una secuencia representada por un vector $d = (d_0, ..., d_{l-1})$ con un único valor diferente de cero, donde la *i-ésima* posición es asignado al *i-ésimo* usuario/grupo. El *i-ésimo* índice con valor diferente a cero recibe un valor $\beta$, $d_i = \beta$, donde $\beta$ es el peso de la marca.

2. Después, se realiza una DCT inversa (IDCT, por sus siglas en Inglés) obteniendo una señal coseno.

3. La señal es multiplicada por una secuencia de ruido pseudoaleatorio *PN* con valores -1 y 1. La secuencia de espectro disperso asignada para el i-ésimo usuario/grupo es definida como:

$$w_i = PN(s) \otimes DCT(i, \beta) \tag{4.1}$$

donde: *PN(s)* es una secuencia pseudoaleatoria generada usando una llave *s*, $DCT(i, \beta)$ es el *i-ésimo* vector base DCT de una *l-tupla* de fuerza $\beta$ y $\otimes$ implica una multiplicación punto a punto. En la Figura 4.2 se ilustra la generación del identificador.

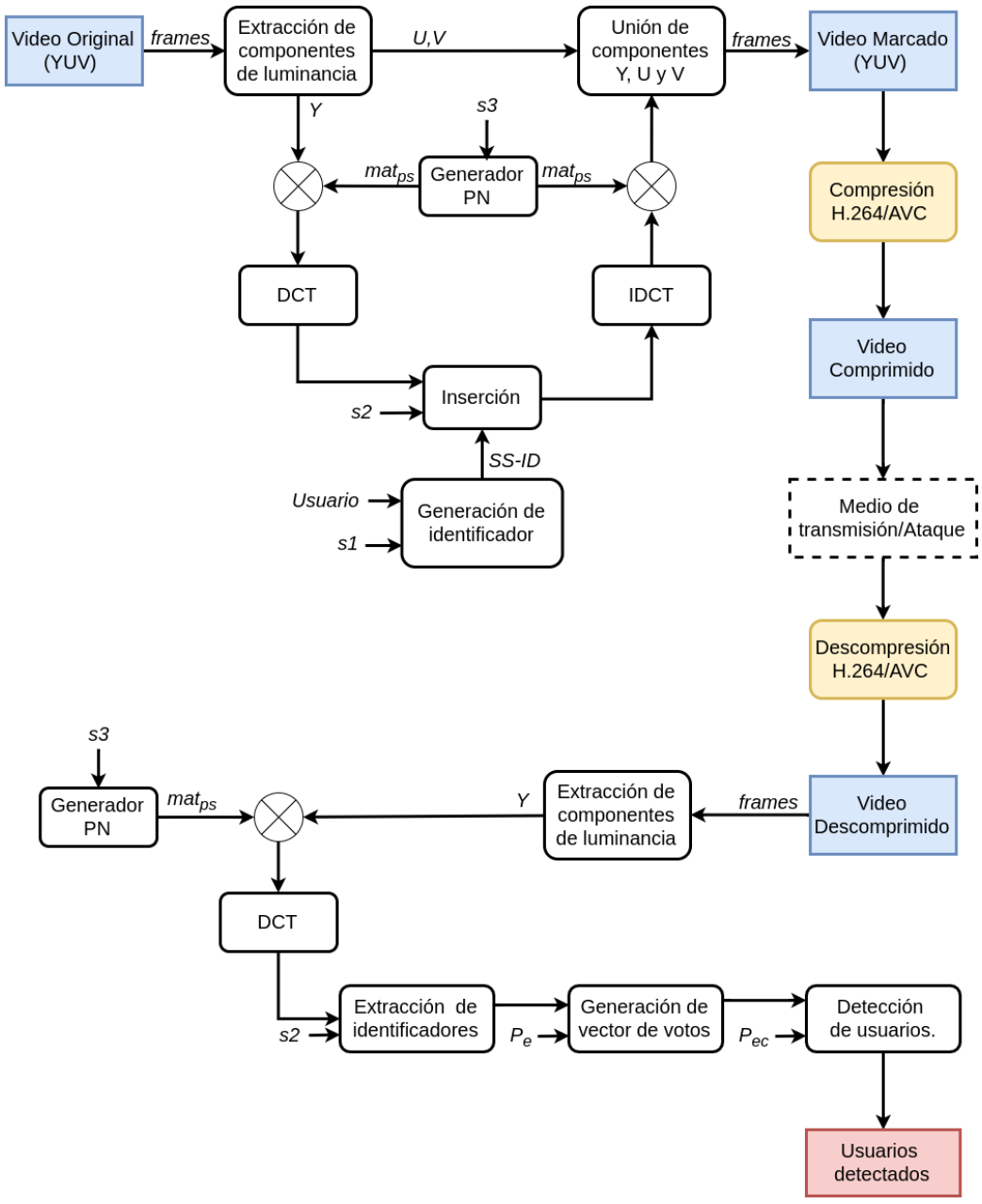


Figura 4.1: Diagrama completo del método propuesto.

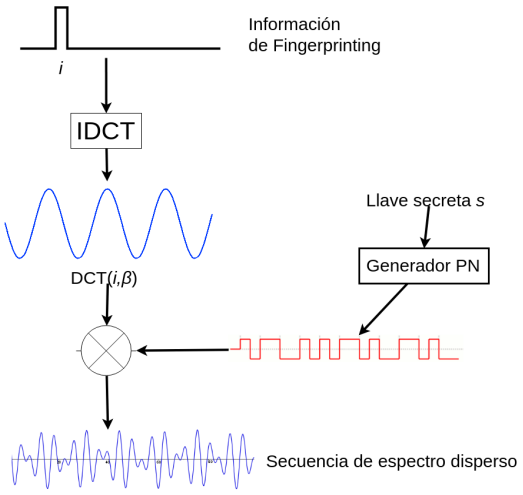


Figura 4.2: Proceso de generación de identificador.

Para generar la secuencia de espectro correspondiente al grupo (ID de grupo) se utiliza un valor *s* para generar el ruido pseudoaleatorio PN(*s*), mientras que para generar la secuencia de espectro correspondiente al usuario (ID de usuario) se usa el número del grupo al que pertenece para generar el ruido pseudoaleatorio $PN(IDdeGrupo)$.

### 4.1.2 Inserción del identificador

La técnica de marcado de agua utiliza una técnica de espectro disperso en el dominio de la frecuencia mediante la DCT. Para este trabajo se proponen 3 enfoques de inserción donde la diferencia radica en el tamaño de bloque y el número de coeficientes seleccionados (véase la Sección 4.2 ). De manera general para insertar la secuencia de espectro del identificador se realiza el siguiente proceso para cada *frame*:

1. Primero, atendiendo a la contra medida presentada en [29] el *frame* es transformado a sus componentes de luminancia (Y) y crominancia (Cb, Cr). El componente de luminancia, que es donde es insertado el identificador, es multiplicado por una matriz pseudoaleatoria $mat_{ps}$ con distribución normal de valores 1 y -1.

2. Se convierte los componentes Y, o en su defecto bloque de componentes Y, del dominio espacial al frecuencial mediante la DCT.

3. Se seleccionan $l_g + l_u$ coeficientes DCT de baja y media frecuencia en base a una llave $k$, donde $l_g$ es la longitud del ID de grupo y $l_u$ la del ID de usuario. Los coeficientes seleccionados se denotan como $v_g = (v_0, ..., v_{lg-1})$, $v_u = (v_l g, ..., v_{(lg+lu-1)})$

4. Se generan dos secuencias de espectro $w_{ig}$ y $w_{iu}$ usando una llave $s$, la fuerza de marcado $\beta_g$ y $\beta_u$, de tal modo que para *j-ésimo* usuario del *i-ésimo* grupo:

$$w_{ig} = PN(s) \otimes DCT(i, \beta_g) \tag{4.2}$$

$$w_{iu} = PN(i) \otimes DCT(j, \beta_u) \tag{4.3}$$

5. Para insertar el identificador se insertan las secuencias de espectro generadas $w_{ig}$ y $w_{iu}$ en $v_g$ y $v_u$, es decir:

$$v_g* = v_g + w_{ig} \tag{4.4}$$

$$v_u* = v_u + w_{iu} \tag{4.5}$$

6. Se transforman los coeficientes DCT del dominio frecuencial al espacial con la transformada inversa del coseno (IDCT).

7. Los componentes de luminancia son multiplicados por la matriz pseudoaleatoria $mat_{ps}$. Finalmente, los componentes de luminancia y crominancia son convertidos al espacio de colores RGB.

### 4.1.3 Detección del identificador

Cabe mencionar que para la detección de los identificadores de los usuarios que participaron en el ataque se detecta primero el grupo (ID de grupo) y posteriormente el usuario (ID de usuario). Si bien, la detección puede realizar sólo con la copia del video para obtener una mejor detección se hace uso del video original. De igual forma que la inserción la extracción es *frame* por *frame*. Para la detección se realizan los siguientes pasos:

1. Se obtienen los componentes de luminancia de cada *frame* tanto del video pirata como el video original, se obtiene una diferencia restando los componentes del video original a los componentes del video pirata. Esta diferencia se multiplican por la matriz pseudoaleatoria $mat_{ps}$ utilizada en el proceso de inserción.

2. Se convierte los componentes de luminancia al dominio de la frecuencia mediante la DCT.

3. Se selecciona aleatoriamente con la llave *k* los $l_g$ y $l_u$ coeficientes de las frecuencias bajas y medias denotadas como $\widetilde{v}_g$ y $\widetilde{v}_u$.

4. Para la detección de ID de grupo.

   *a*) Se genera la secuencia *PN* utilizando la llave *s*

   *b*) Se realiza una DCT para obtener la secuencia de detección $d_g$

$$\widetilde{d}_g = DCT(PN(s) \otimes (\widetilde{v}_g - v_g)) \tag{4.6}$$

   *c*) Ya que el resultado es una señal que contiene ruido es necesario determinar un umbral $T_g$ con una probabilidad de falso-positivo $Pe_g$ el cual define que usuarios participaron en el ataque.

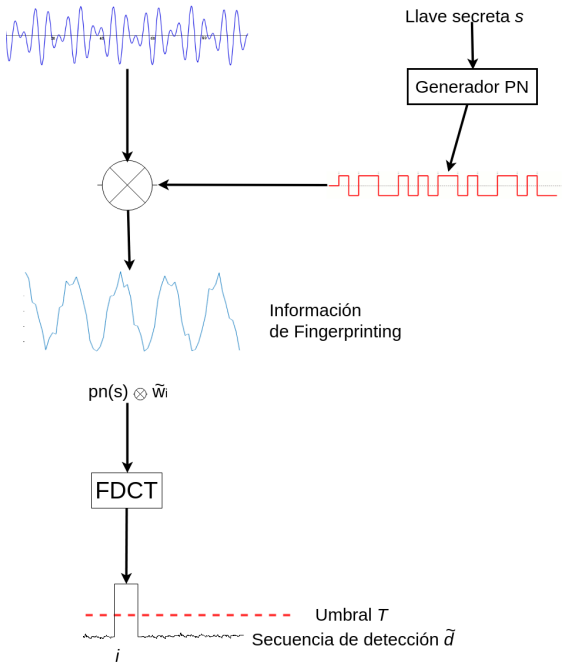


Figura 4.3: Proceso de detección de identificador.

*d*) Si $\widetilde{d}_{g,k} \geq T_g$, $(0 \leq k \leq l-1)$ se determina *k* como ID de grupo.

5. Para detectar los ID de usuarios es necesario realizar este proceso por cada uno de los grupos detectados.

   *a*) Se genera la secuencias *PN* utilizando la llave ID de grupo $ID_g$

   *b*) Se realiza una DCT para obtener la detección de la secuencia $d_u$

$$\widetilde{d}_u = DCT(PN(ID_g) \otimes (\widetilde{v}_u - v_u)) \tag{4.7}$$

   *c*) De igual forma que en la detección de grupos se determina un umbral $T_u$ con una probabilidad de falso-positivo $Pe_u$

   *d*) Si $\widetilde{d}_{u,k} \geq T_u$, $(0 \leq k \leq l-1)$ se determina *k* como ID de usuario.

En la Figura 4.3 se muestra el proceso de detección de identificador.

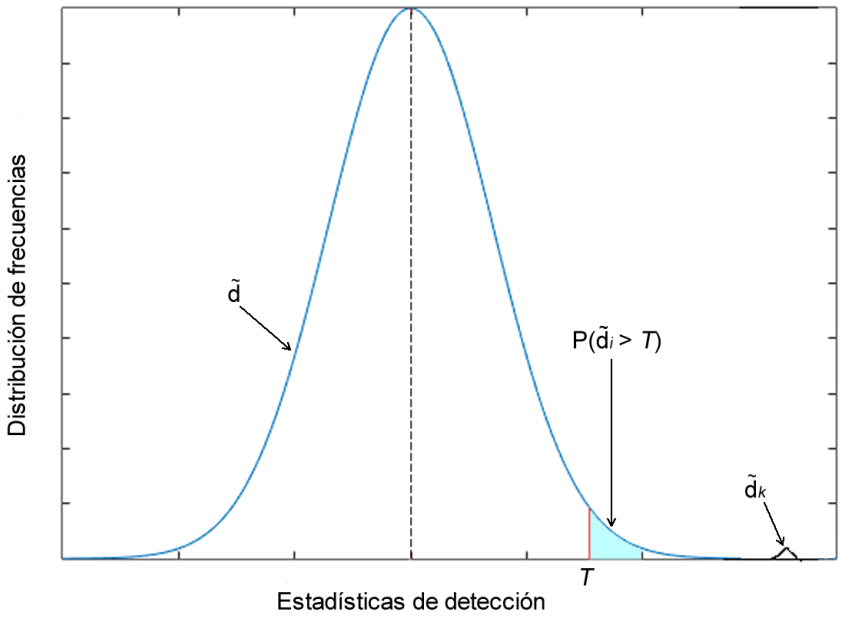


Figura 4.4: Diagrama de la operación de dispersión.

### 4.1.4 Cálculo de umbral para detección del identificador

Como se mencionó en la Sección 4.1.3, se hace uso de un umbral el cual separa el ruido generado y la información del identificador. Para establecer el valor de dicho umbral $T$ se utiliza la distribución de la secuencia de detección $\widetilde{d}$, la cual se puede modelar como una distribución gaussiana con media cero descartando la información del identificador $\widetilde{d}_k$ como se muestra en la Figura 4.4. Sin embargo, existe la posibilidad de que información del identificador se encuentre dentro de la distribución gaussiana por lo que es necesario considerar una probabilidad de error $p_e$.

De tal forma que el valor del umbral se calcula como:

$$T = \sqrt{2\sigma_g^2} erfc^{-1}(2Pe_g) \tag{4.8}$$

donde:

$$\sigma_g^2 = \frac{1}{n} \sum \widetilde{d}_{g,k \in D_g} \left( \widetilde{d}_{g,k} - \overline{\widetilde{d}_g} \right)^2 \tag{4.9}$$

y

$erfc^{-1}(.)$ corresponde a la función de error complementaria inversa.

## 4.2 Adaptación del método *fingerprinting* al estándar H.264/AVC

Para la adaptación del método [28] con el estándar H.264/AVC se plantean tres propuestas para determinar cuál de ellas brinda mejores resultados.
En las tres propuestas el identificador es insertado en los coeficientes DCT de los componentes de luminancia de los *frames* del video sin compresión, mientras que la detección del identificador se realiza en los coeficientes DCT de los *frames* obtenidos después de la compresión. Sin embargo, considerando que los identificadores ya insertados pasarán por el proceso completo de compresión con pérdida del estándar H.264/AVC se puede generar pérdidas significativas a la información del identificador. Se apuesta por esta propuesta debido a que la técnica empleada en [28] es una propuesta enfocada a una compresión con pérdida para imágenes (JPEG) por lo que es altamente posible recuperar la información del identificador aún después de la compresión con perdida del compresor H.264/AVC. En la Figura 4.5 se muestra el lugar de inserción y detección durante el proceso de compresión y descompresión del estándar H.264/AVC.

La primer propuesta es diseñada para perfiles *Baseline*, la segunda para a perfiles *High* y la tercera aplica para ambos perfiles. La diferencia entre las tres propuestas radica en el tamaño de bloque y el número de coeficiente que se toma por cada bloque para hacer la inserción y detección del identificador. Para las dos primeras propuestas la inserción del identificador no se realiza en la imagen completa, sino en bloques de píxeles de diferente tamaño. En cada uno de los bloques parte del identificador es insertado utilizando la misma técnica de espectro disperso. El tamaño del bloque se definirá según las características del perfil, las tres adaptaciones se describen a continuación.

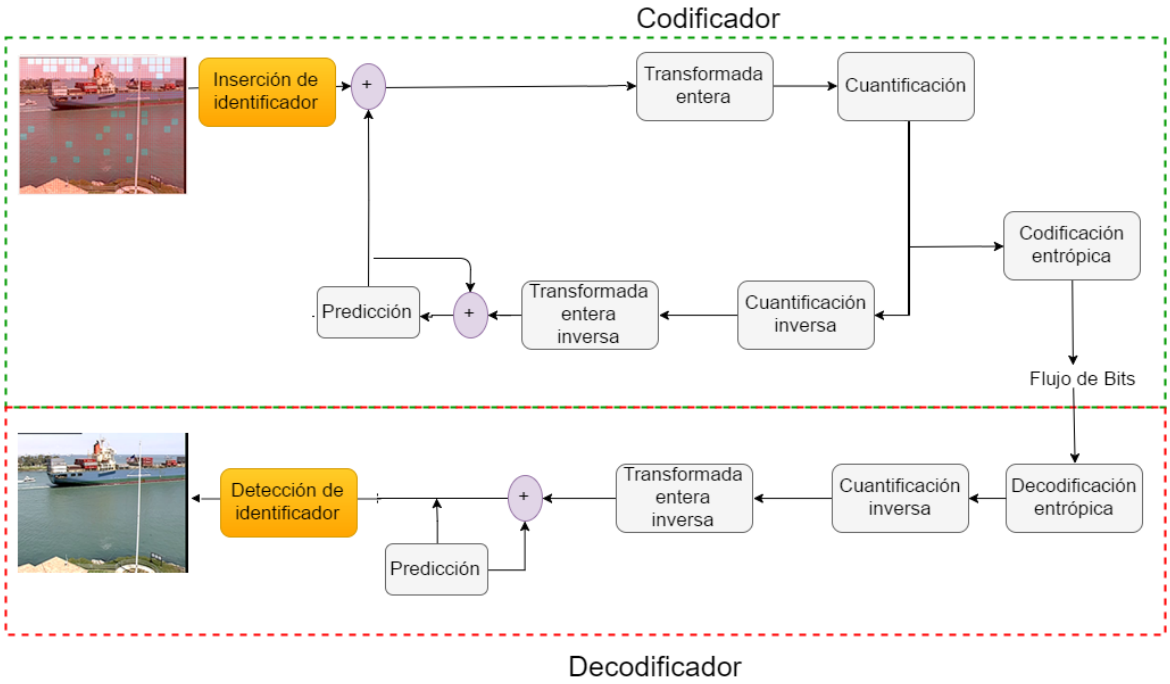


Figura 4.5: Inserción y detección del identificador en el proceso del estándar H.264/AVC.

### 4.2.1 Diseño para perfiles *Baseline*

Teniendo en cuenta que para perfiles *Baseline* se utilizan bloques de tamaño $4 \times 4$ y $16 \times 16$ (predicción Intra) se utilizan transformadas de $4 \times 4$, aún para el caso de bloques $16 \times 16$ ya que en este último se divide en subbloques de $4 \times 4$. Por tal motivo la inserción del identificador se realiza en bloques de $4 \times 4$. Se utiliza la transformada entera $4 \times 4$ utilizada en H.264/AVC. Por cada bloque de $4 \times 4$ se escoge aleatoriamente un coeficiente de bajas y medias frecuencias donde se inserta un elemento del identificador, de esta forma se seleccionan 16 coeficientes por cada macrobloque. La razón por que se escogen bajas y medias frecuencias es que éstas son resistentes a la perdida de información generada por una compresión con pérdida. Dado que las frecuencias altas representan información que no es de gran perceptibilidad al ojo humano la eliminación de estas frecuencias produce menos perdida de calidad que al eliminar bajas o medias frecuencias. Además, se seleccionan sólo algunos de los coeficientes aleatoriamente debido a que proporciona seguridad. Si se usaran todos los coeficientes pertenecientes a las bajas y medias frecuencias el atacante sabría

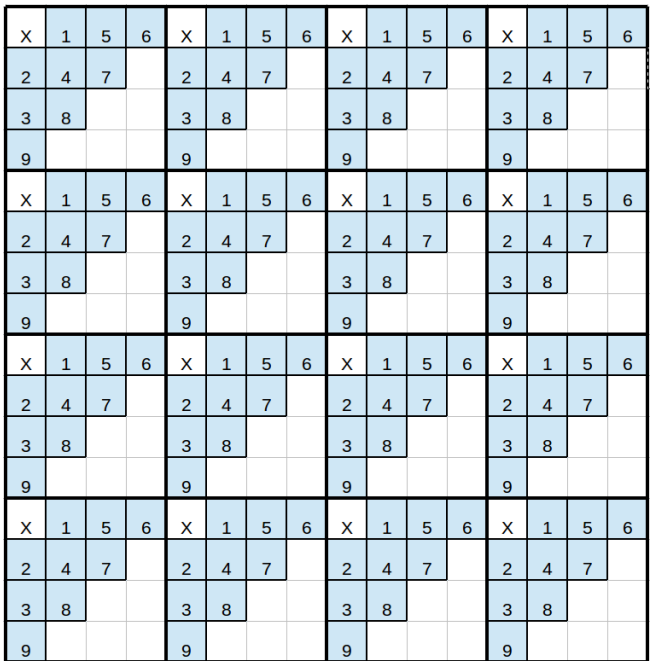


Figura 4.6: Coeficientes disponibles para la inserción en un macrobloque con bloques de $4 \times 4$.

con exactitud que coeficientes modificar para distorsionar el identificador, al ser seleccionado sólo algunos y de manera aleatoria no se sabe con precisión que coeficientes atacar. En la Figura 4.6 se muestran los coeficientes disponibles para realizar la inserción en un macrobloque con bloques de $4 \times 4$.

### 4.2.2 Diseño para perfiles *High*

Para el caso del perfil *High*, en el cual se introducen tanto predicciones y transformadas de $8 \times 8$, la inserción se realiza en bloques de $8 \times 8$ como se describe en 4.2.1 y además en bloques de $8 \times 8$. Se utiliza la transformada entera $8 \times 8$ utilizada en H.264/AVC. En este último tipo de bloques se seleccionan cuatro coeficientes, de igual manera que en la inserción en $4 \times 4$, de baja y media frecuencia aleatoriamente. De esta forma se seleccionan, al igual que en los macrobloques con bloques de $4 \times 4$, 16 coeficientes por macrobloques. En la Figura 4.7 se muestran los coeficientes disponibles para realizar la inserción en un macrobloque con bloques de $8 \times 8$.

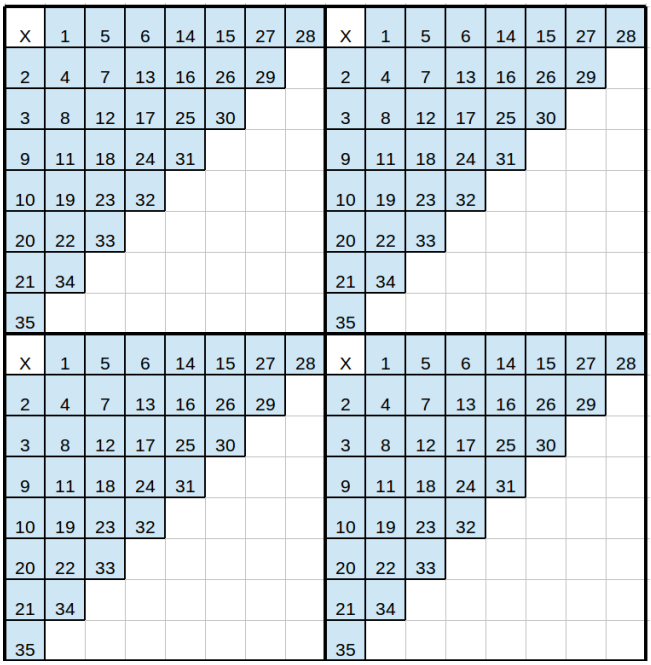


Figura 4.7: Coeficientes disponibles para la inserción en un macrobloque con bloques de $8 \times 8$.

### 4.2.3 Diseño para inserción en *frame* completo

La tercera propuesta se basa en no obtener los coeficientes DCT por bloques para la inserción, sino obtenerlos del *frame* completo, es decir, aplicar la trasformada a todo el *frame*. Para este caso se utiliza la DCT debido a que si bien es costosa, a diferencia de las técnicas de inserción anteriores, sólo se realiza una vez por *frame*. Del total de los coeficientes DCT del *frame* se toman $l_g + l_u$ coeficientes aleatoriamente y sean pertenecientes a bajas y medias frecuencias. Donde $l_g$= longitud del ID de grupo y $l_u$= longitud del ID de usuario. Para un *frame* de $32 \times 32$ píxeles, en la Figura 4.8 se muestra los coeficientes disponibles para realizar la inserción en un *frame* completo.

### 4.2.4 Enfoque de conteo de votos de identificadores

Dado que el identificador se puede insertar más de una vez (uno por *frame*) y tomando en cuenta los diferentes tipos de *frames* que existen en el estándar H.264/AVC, la detección de cada uno de los identificadores detectados puede dar un resultado diferente. Para hacer frente a esta condición

| x | 1 | 5 | 6 | 14 | 15 | 27 | 28 | 44 | 45 | 65 | 66 | 90 | 91 | 119 | 120 | 152 | 153 | 189 | 190 | 230 | 231 | 275 | 276 | 324 | 325 | 377 | 378 | 434 | 435 | 495 | 496 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 4 | 7 | 13 | 16 | 26 | 29 | 43 | 46 | 64 | 67 | 89 | 92 | 118 | 121 | 151 | 154 | 188 | 191 | 229 | 232 | 274 | 277 | 323 | 326 | 376 | 379 | 433 | 436 | 494 | 497 | 558 |
| 3 | 8 | 12 | 17 | 25 | 30 | 42 | 47 | 63 | 68 | 88 | 93 | 117 | 122 | 150 | 155 | 187 | 192 | 228 | 233 | 273 | 278 | 322 | 327 | 375 | 380 | 432 | 437 | 493 | 498 | 557 | 559 |
| 9 | 11 | 18 | 24 | 31 | 41 | 48 | 62 | 69 | 87 | 94 | 116 | 123 | 149 | 156 | 186 | 193 | 227 | 234 | 272 | 279 | 321 | 328 | 374 | 381 | 431 | 438 | 492 | 499 | 556 | 560 | 617 |
| 10 | 19 | 23 | 32 | 40 | 49 | 61 | 70 | 86 | 95 | 115 | 124 | 148 | 157 | 185 | 194 | 226 | 235 | 271 | 280 | 320 | 329 | 373 | 382 | 430 | 439 | 491 | 500 | 555 | 561 | 616 | 618 |
| 20 | 22 | 33 | 39 | 50 | 60 | 71 | 85 | 96 | 114 | 125 | 147 | 158 | 184 | 195 | 225 | 236 | 270 | 281 | 319 | 330 | 372 | 383 | 429 | 440 | 490 | 501 | 554 | 562 | 615 | 619 | 672 |
| 21 | 34 | 38 | 51 | 59 | 72 | 84 | 97 | 113 | 126 | 146 | 159 | 183 | 196 | 224 | 237 | 269 | 282 | 318 | 331 | 371 | 384 | 428 | 441 | 489 | 502 | 553 | 563 | 614 | 620 | 671 | 673 |
| 35 | 37 | 52 | 58 | 73 | 83 | 98 | 112 | 127 | 145 | 160 | 182 | 197 | 223 | 238 | 268 | 283 | 317 | 332 | 370 | 385 | 427 | 442 | 488 | 503 | 552 | 564 | 613 | 621 | 670 | 674 | 723 |
| 36 | 53 | 57 | 74 | 82 | 99 | 111 | 128 | 144 | 161 | 181 | 198 | 222 | 239 | 267 | 284 | 316 | 333 | 369 | 386 | 426 | 443 | 487 | 504 | 551 | 565 | 612 | 622 | 669 | 675 | 722 | 724 |
| 54 | 56 | 75 | 81 | 100 | 110 | 129 | 143 | 162 | 180 | 199 | 221 | 240 | 266 | 285 | 315 | 334 | 368 | 387 | 425 | 444 | 486 | 505 | 550 | 566 | 611 | 623 | 668 | 676 | 721 | 725 | 770 |
| 55 | 76 | 80 | 101 | 109 | 130 | 142 | 163 | 179 | 200 | 220 | 241 | 265 | 286 | 314 | 335 | 367 | 388 | 424 | 445 | 485 | 506 | 549 | 567 | 610 | 624 | 667 | 677 | 720 | 726 | 769 | 771 |
| 77 | 79 | 102 | 108 | 131 | 141 | 164 | 178 | 201 | 219 | 242 | 264 | 287 | 313 | 336 | 366 | 389 | 423 | 446 | 484 | 507 | 548 | 568 | 609 | 625 | 666 | 678 | 719 | 727 | 768 | 772 | 813 |
| 78 | 103 | 107 | 132 | 140 | 165 | 177 | 202 | 218 | 243 | 263 | 288 | 312 | 337 | 365 | 390 | 422 | 447 | 483 | 508 | 547 | 569 | 608 | 626 | 665 | 679 | 718 | 728 | 767 | 773 | 812 | 814 |
| 104 | 106 | 133 | 139 | 166 | 176 | 203 | 217 | 244 | 262 | 289 | 311 | 338 | 364 | 391 | 421 | 448 | 482 | 509 | 546 | 570 | 607 | 627 | 664 | 680 | 717 | 729 | 766 | 774 | 811 | 815 | 852 |
| 105 | 134 | 138 | 167 | 175 | 204 | 216 | 245 | 261 | 290 | 310 | 339 | 363 | 392 | 420 | 449 | 481 | 510 | 545 | 571 | 606 | 628 | 663 | 681 | 716 | 730 | 765 | 775 | 810 | 816 | 851 | 853 |
| 135 | 137 | 168 | 174 | 205 | 215 | 246 | 260 | 291 | 309 | 340 | 362 | 393 | 419 | 450 | 480 | 511 | 544 | 572 | 605 | 629 | 662 | 682 | 715 | 731 | 764 | 776 | 809 | 817 | 850 | 854 | 887 |
| 136 | 169 | 173 | 206 | 214 | 247 | 259 | 292 | 308 | 341 | 361 | 394 | 418 | 451 | 479 | 512 | 543 | 573 | 604 | 630 | 661 | 683 | 714 | 732 | 763 | 777 | 808 | 818 | 849 | 855 | 886 | 888 |
| 170 | 172 | 207 | 213 | 248 | 258 | 293 | 307 | 342 | 360 | 395 | 417 | 452 | 478 | 513 | 542 | 574 | 603 | 631 | 660 | 684 | 713 | 733 | 762 | 778 | 807 | 819 | 848 | 856 | 885 | 889 | 918 |
| 171 | 208 | 212 | 249 | 257 | 294 | 306 | 343 | 359 | 396 | 416 | 453 | 477 | 514 | 541 | 575 | 602 | 632 | 659 | 685 | 712 | 734 | 761 | 779 | 806 | 820 | 847 | 857 | 884 | 890 | 917 | 919 |
| 209 | 211 | 250 | 256 | 295 | 305 | 344 | 358 | 397 | 415 | 454 | 476 | 515 | 540 | 576 | 601 | 633 | 658 | 686 | 711 | 735 | 760 | 780 | 805 | 821 | 846 | 858 | 883 | 891 | 916 | 920 | 945 |
| 210 | 251 | 255 | 296 | 304 | 345 | 357 | 398 | 414 | 455 | 475 | 516 | 539 | 577 | 600 | 634 | 657 | 687 | 710 | 736 | 759 | 781 | 804 | 822 | 845 | 859 | 882 | 892 | 915 | 921 | 944 | 946 |
| 252 | 254 | 297 | 303 | 346 | 356 | 399 | 413 | 456 | 474 | 517 | 538 | 578 | 599 | 635 | 656 | 688 | 709 | 737 | 758 | 782 | 803 | 823 | 844 | 860 | 881 | 893 | 914 | 922 | 943 | 947 | 968 |
| 253 | 298 | 302 | 347 | 355 | 400 | 412 | 457 | 473 | 518 | 537 | 579 | 598 | 636 | 655 | 689 | 708 | 738 | 757 | 783 | 802 | 824 | 843 | 861 | 880 | 894 | 913 | 923 | 942 | 948 | 967 | 969 |
| 299 | 301 | 348 | 354 | 401 | 411 | 458 | 472 | 519 | 536 | 580 | 597 | 637 | 654 | 690 | 707 | 739 | 756 | 784 | 801 | 825 | 842 | 862 | 879 | 895 | 912 | 924 | 941 | 949 | 966 | 970 | 987 |
| 300 | 349 | 353 | 402 | 410 | 459 | 471 | 520 | 535 | 581 | 596 | 638 | 653 | 691 | 706 | 740 | 755 | 785 | 800 | 826 | 841 | 863 | 878 | 896 | 911 | 925 | 940 | 950 | 965 | 971 | 986 | 988 |
| 350 | 352 | 403 | 409 | 460 | 470 | 521 | 534 | 582 | 595 | 639 | 652 | 692 | 705 | 741 | 754 | 786 | 799 | 827 | 840 | 864 | 877 | 897 | 910 | 926 | 939 | 951 | 964 | 972 | 985 | 989 | 1002 |
| 351 | 404 | 408 | 461 | 469 | 522 | 533 | 583 | 594 | 640 | 651 | 693 | 704 | 742 | 753 | 787 | 798 | 828 | 839 | 865 | 876 | 898 | 909 | 927 | 938 | 952 | 963 | 973 | 984 | 990 | 1001 | 1003 |
| 405 | 407 | 462 | 468 | 523 | 532 | 584 | 593 | 641 | 650 | 694 | 703 | 743 | 752 | 788 | 797 | 829 | 838 | 866 | 875 | 899 | 908 | 928 | 937 | 953 | 962 | 974 | 983 | 991 | 1000 | 1004 | 1013 |
| 406 | 463 | 467 | 524 | 531 | 585 | 592 | 642 | 649 | 695 | 702 | 744 | 751 | 789 | 796 | 830 | 837 | 867 | 874 | 900 | 907 | 929 | 936 | 954 | 961 | 975 | 982 | 992 | 999 | 1005 | 1012 | 1014 |
| 464 | 466 | 525 | 530 | 586 | 591 | 643 | 648 | 696 | 701 | 745 | 750 | 790 | 795 | 831 | 836 | 868 | 873 | 901 | 906 | 930 | 935 | 955 | 960 | 976 | 981 | 993 | 998 | 1006 | 1011 | 1015 | 1020 |
| 465 | 526 | 529 | 587 | 590 | 644 | 647 | 697 | 700 | 746 | 749 | 791 | 794 | 832 | 835 | 869 | 872 | 902 | 905 | 931 | 934 | 956 | 959 | 977 | 980 | 994 | 997 | 1007 | 1010 | 1016 | 1019 | 1021 |
| 527 | 528 | 588 | 589 | 645 | 646 | 698 | 699 | 747 | 748 | 792 | 793 | 833 | 834 | 870 | 871 | 903 | 904 | 932 | 933 | 957 | 958 | 978 | 979 | 995 | 996 | 1008 | 1009 | 1017 | 1018 | 1022 | 1023 |

Figura 4.8: Coeficientes disponibles para la inserción por *frame* completo.

se propone considerar un sistema de votos donde se toma en cuenta cada una de las detecciones de usuarios realizadas. De esta forma cada detección por *frame* da un resultado preliminar de la detección de los usuarios coludidos entre sí. Considerando que si la probabilidad de error ($p_e$) es lo bastante pequeña para no detectar falsos-positivos (usuarios inocentes acusados) se genera un filtro muy riguroso lo cual provoca una detección baja de usuarios coludidos entre sí. Por el contrario, se observó en pruebas preliminares que la implementación de un umbral no tan riguroso permite una mayor detección de coludidos, sin embargo, provoca que se detecten algunos falsos-positivo. Por este motivo se propone la implementación de un segundo umbral con una probabilidad de error ($p_{ec}$) que permitirá que dichos falsos-positivos se descarten y a diferencia de utilizar un solo umbral será mayor la detección de usuarios coludidos entre sí.

### 4.2.5 Definición de umbral para el sistema de conteo de votos

Debido a que cada identificador, aplicando en el primer umbral $T$, tendrá un resultado preliminar en el cual es posible que existan falsos-positivos, se genera un vector de votaciones ($V_c$) el cual es la suma de todos los resultados preliminares. En dicho vector de votaciones se tendrán que aplicar el segundo umbral $T_c$ para eliminar todos los falsos-positivos.

Dado que el vector de conteos sólo son valores positivos el límite inferior de la distribución siempre será cero. Con las pruebas preliminares se pudo determinar que la distribución de $V_c$ se puede modelar como una distribución media gaussiana la cual es definida como [15]:

$$f(y;\sigma) = \sqrt{\frac{2}{\sigma^2\pi}} exp\left(-\frac{y^2}{2\sigma^2}\right) y > 0 \tag{4.10}$$

con una función de distribución acumulativa definida como:

$$F(y;\sigma) = \int_0^y \sqrt{\frac{2}{\sigma^2\pi}} exp\left(-\frac{x^2}{2\sigma^2}\right) dx \tag{4.11}$$

Para un umbral $T_c$ dado la probabilidad de detección falsa $Pe_c$ es calculada por la resta de la función de distribución acumulativa a la unidad definida como:

$$Pe_c = 1 - \int_0^{T_c} \sqrt{\frac{2}{\sigma^2\pi}} exp\left(-\frac{y^2}{2\sigma^2}\right) dy \tag{4.12}$$

Usando un cambio de variable $Z = y/(\sqrt{2\sigma^2})$ en la ecuación (4.12) teniendo:

$$F(y;\sigma) = \frac{2}{\sqrt{\pi}} \int_0^{y/(\sqrt{2\sigma^2})} exp(-z^2)dz = erf\left(\frac{y}{\sqrt{2\sigma^2}}\right) \tag{4.13}$$

donde $erf(\cdot)$ es la función de error y está relacionada a la función de error complementaria $erfc(\cdot)$ como:

$$erfc(x) = 1 - erf(x) \tag{4.14}$$

de las ecuaciones (4.12), (4.13) y (4.14) la probabilidad de detección falsa $Pe_c$ para un umbral $T_c$ puede ser expresado como:

$$P_{ec} = erfc\left(\frac{T_c}{\sqrt{2\sigma^2}}\right) \tag{4.15}$$

Por lo tanto el segundo umbral para la detección de grupos $T_{cg}$ para una probabilidad de detección errónea de grupo $Pec_g$ se calcula como:

$$T_{cg} = \sqrt{2\sigma_{Cg}^2} erfc^{-1}(Pec_g) \tag{4.16}$$

donde $\sigma_{Cg}^2$= la varianza del vector de conteo de grupos $C_g$.

De igual forma que se calcula el segundo umbral para la detección de usuarios $T_{cu}$ para una probabilidad de detección errónea de usuarios $Pec_u$ se calculado como:

$$T_{cu} = \sqrt{2\sigma_{Cu}^2} erfc^{-1}(Pec_u) \tag{4.17}$$

donde $\sigma_{Cu}^2$= la varianza del vector de conteo de usuarios $C_u$

## 4.3 Resumen

En este capítulo se describió las características de la técnica de marcado de agua y el identificador utilizado. También se describió la adaptación de la método *fingerprinting* al estándar H.264/AVC. Así mismo, se detalló la forma en que se genera, inserta y se extrae el identificador. De manera particular se explicó las propuestas planteadas para el perfil *Main*, para el perfil *High* y una tercera propuesta para ambos perfiles. Dado que la inserción se hace *frame* por *frame* se considera un resultado preliminar por cada identificador extraído, creando así un sistema de conteo. Para obtener el mayor desempeño se aplica un umbral al sistema de conteo.

# 5

# Experimentación y resultados

Es este capítulo se describen las experimentaciones realizadas así como los resultados obtenidos para cada una de las tres propuestas de inserción del identificador. Se describen los cuatro módulos creados: generador e inserción de identificador, compresor y descompresor, generador de ataques y detección de identificadores. Además, se presenta el escenario bajo el cual se realizan las pruebas. Las pruebas realizadas evalúan la calidad del video marcado, bajo las métricas de PSNR y QNR, y la robustez que presenta bajo diferentes ataques de colusión.

## 5.1 Infraestructura utilizada

Las pruebas realizadas se llevaron a cabo en equipo de cómputo portátil con procesador i7 a 2.3 GHz, 12 GB de memoria RAM, disco duro de 1 TB y un sistema operativo Ubuntu 14.04 64 bits. Se hizo uso del software MatLab para la implementación del generador, del método de inserción y el detector de identificadores. También se utilizó el software de referencia del estándar H.264/AVC denominado JM, debido a que dicho software está implementado en el lenguaje de codificación C se

utilizó el compilador de lenguaje C GNU Compiler Collection.

## 5.2 Generación de módulos del método propuesto

El primer módulo está compuesto por la generación e inserción del identificador, por lo que es implementado el método *fingerprinting* descrito en el Capítulo 4. Un segundo módulo es creado en el cual se genera una compresión seguida de una descompresión mediante el uso de la herramienta JM, el cual es un software de referencia del estándar H.264/AVC, en donde se definen las características de compresión como el perfil y valor QP de compresión. El tercer módulo es el encargado de generar los ataques de colusión. En este módulo $nc$ videos son necesarios para generar un ataque de $nc$ usuarios coludidos entre sí. Los ataques de colusión posibles son: promedio, mínimo, máximo, mediana y minmax. El cuarto módulo es el encargado de detectar los identificadores y a su vez determinar que usuarios participaron en el ataque de colusión.

El proceso de prueba inicia con la generación e inserción del identificador a un video sin compresión (formato YUV). Después el video es comprimido, (teniendo como resultado un video en formato H.264/AVC) y a su vez descomprimido. Al video le es aplicado un ataque de colusión específico, por lo que se hace uso de diferentes videos pertenecientes a un conjunto de usuarios. Finalmente, se extraen los identificadores del video resultante del ataque, con esto se determina cuáles son los usuarios deshonestos.

En la Figura 5.1 se observa el proceso completo de una prueba.

## 5.3 Descripción de los experimentos

Con las pruebas realizadas se desea tener una evaluación de las características de calidad, capacidad y robustez, para el método propuesto. Para la experimentación se utilizan un conjunto de 30 videos sin compresión (en formato YUV) de resolución CIF ($352 \times 288$ píxeles) que comúnmente son

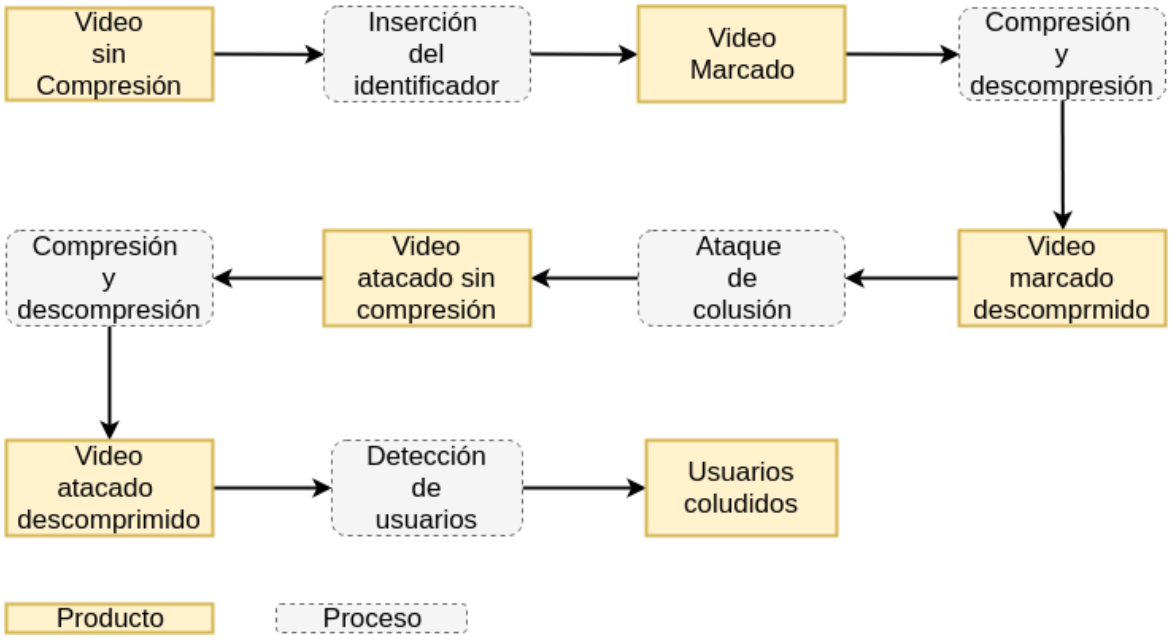


Figura 5.1: Diagrama del proceso de pruebas.

usados en la comunidad científica debido a que presentan diferentes combinaciones de movimiento, color, contraste y objetos. Los videos están disponibles en [57].

En cuanto a los parámetros de compresión, se utiliza un valor QP=18 debido a que este parámetro es utilizado en [42, 46] y se desea tener una comparación de robustez directamente con estos trabajos. Con respecto al perfil de compresión se elige según la técnica de inserción:

- Las pruebas realizadas usando la técnica de inserción mediante bloques de $4 \times 4$ se comprimen con un perfil *Baseline* debido a que en dicho perfil se utilizan transformadas y predicciones con bloques de tamaño $4 \times 4$.

- Las pruebas realizadas usando la técnica de inserción mediante bloques de $8 \times 8$ se comprimen con un perfil *High* debido a que en dicho perfil se utilizan transformadas y predicciones con bloques de $8 \times 8$.

- Las pruebas realizadas usando la técnica de inserción mediante *frame* completo se comprimen con un perfil tanto *Baseline* como *High*.

## 5.4 Pruebas de calidad visual

La calidad del video marcado está definida por la longitud del identificador y la energía de la marca la cual está dada por:

$$\beta^2 = \beta_g^2 + \beta_u^2 \tag{5.1}$$

### 5.4.1 Escenario de prueba

Para fines de evaluación de calidad se realizan pruebas de inserción para distintos valores de $\beta_g$ y $\beta_u$ con una longitud fija de $l_g = 1024$ y $l_u = 1024$. Cabe mencionar que mientras más grande sean los valores $\beta_g$ y $\beta_u$ mayor será de la detección de usuarios y a su vez menor la calidad del video. Los videos marcados son evaluados bajo las métricas PSNR y QNR. Para los valores PSNR mayores a 37 dB se considera que existe una baja degradación de calidad. Sin embargo, en diferentes trabajos [25, 34, 36, 42, 46] reportan valores mayores a 40 dB, por esta razón se elige una combinación de $\beta_g$ y $\beta_u$ que nos permita obtener valores PSNR mayores a 41.5 dB. Por otra parte, la métrica QNR evalúa la calidad en un rango de 1 a 10, donde 1 es la peor calidad y 10 la mejor. Tomando en cuenta que el valor de calidad fue asignado por una persona y usualmente se considera que en un rango de 1-10 los valores menores a 7 son reprobatorios, para el caso de la métrica QNR se puede considerar que el rango de 1-6 es de mala calidad, los valores entre 7-8 de calidad media y de 9-10 una buena calidad. Así mismo, se toma en cuenta que debido a que habrá un mayor número de usuarios que de grupos asignará un peso mayor al valor $\beta_u$ con respecto al valor $\beta_g$. Siguiendo estas restricciones se elige la combinación de $\beta_g$ y $\beta_u$ que represente una mayor energía, la energía se obtiene a partir de la ecuación 5.1.

Los 30 videos son marcados con cada combinación de $\beta_g$ y $\beta_u$ y se calcula un valor promedio de PSNR y QNR para cada combinación.

| $(\beta_u, \beta_g)$ | 200 | 250 | 300 | 350 | 400 | 450 |
|---|---|---|---|---|---|---|
| 200 | 43.638 | 43.350 | 42.953 | 42.505 | 42.100 | 41.707 |
| 250 | 43.358 | 43.090 | 42.709 | 42.287 | 41.902 | 41.527 |
| 300 | 42.958 | 42.712 | 42.367 | 41.975 | 41.617 | 41.267 |
| 350 | 42.524 | 42.303 | 41.992 | 41.631 | 41.299 | 40.969 |
| 400 | 42.122 | 41.923 | 41.633 | 41.304 | 40.994 | 40.691 |
| 450 | 41.737 | 41.554 | 41.290 | 40.983 | 40.695 | 40.413 |

Tabla 5.1: Resultados de valores PSNR para diferentes combinaciones $\beta_g$ y $\beta_u$ con inserción $4 \times 4$.

## 5.4.2 Resultados de calidad visual

*5.4.2.1. Resultados para la técnica de inserción por bloques* $4 \times 4$

Para seleccionar la combinación más adecuada se obtienen los valores para la métrica PSNR utilizando la técnica de inserción por bloques $4 \times 4$, dichos valores se muestran en la Tabla 5.1. La selección más adecuada se realiza bajo las siguientes restricciones:

1. Primero se selecciona aquellas combinaciones en las que el valor PSNR es mayor a 41.5 dB. En la Figura 5.2 se puede visualizar las combinaciones que pertenece a la región aceptable.

2. Se descartan las combinaciones en las que $\beta_g > \beta_u$.

3. Finalmente, de las combinaciones posibles se calcula la energía de la marca seleccionando la combinación con mayor energía. Para esta técnica de inserción los valores seleccionados son: $\beta_g = 250$ y $\beta_u = 450$.

Por otro lado, en la Tabla 5.3 se presentan los resultados obtenidos de las diferentes combinaciones para la métrica QNR en donde se observa que la combinación seleccionada tiene un valor de 9.577, dicho valor representa una buena calidad.

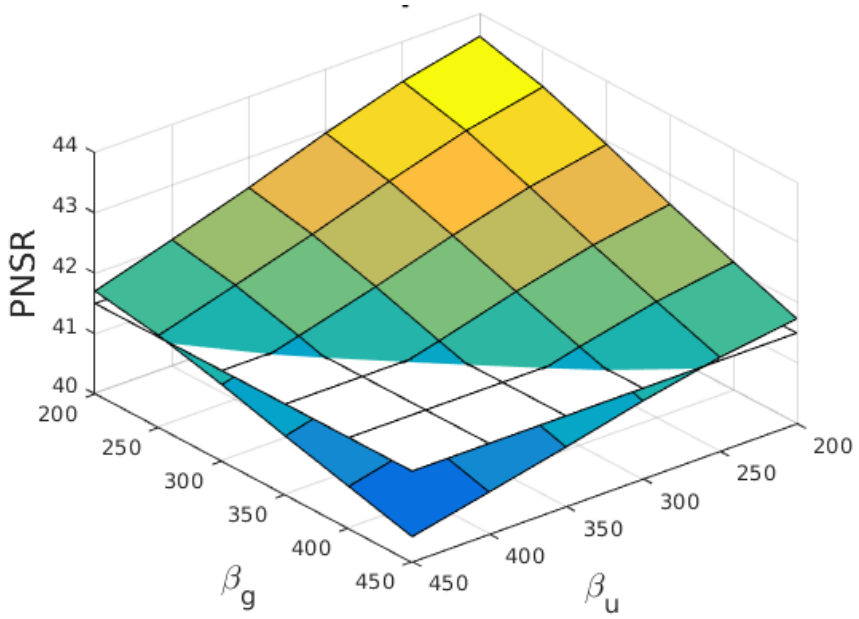


Figura 5.2: Región aceptable para valores de calidad PSNR con inserción $4 \times 4$.

| $(\beta_u, \beta_g)$ | 200 | 250 | 300 | 350 | 400 | 450 |
|---|---|---|---|---|---|---|
| 200 | - | - | - | - | - | - |
| 250 | 320.156 | - | - | - | - | - |
| 300 | 360.555 | 390.512 | - | - | - | - |
| 350 | 403.113 | 430.116 | 460.977 | - | - | - |
| 400 | 447.214 | 471.699 | 500.000 | - | - | - |
| 450 | 492.443 | 514.782 | - | - | - | - |

Tabla 5.2: Energía de las combinación de $\beta_g$ y $\beta_u$ posibles mediante la inserción de bloques de $4 \times 4$.

| $(\beta_g, \beta_u)$ | 200 | 250 | 300 | 350 | 400 | 450 |
|---|---|---|---|---|---|---|
| 200 | 9.771 | 9.749 | 9.715 | 9.678 | 9.642 | 9.608 |
| 250 | 9.749 | 9.725 | 9.691 | 9.656 | 9.620 | 9.586 |
| 300 | 9.713 | 9.691 | 9.657 | 9.620 | 9.586 | 9.553 |
| 350 | 9.671 | 9.650 | 9.618 | 9.583 | 9.549 | 9.516 |
| 400 | 9.632 | 9.612 | 9.579 | 9.546 | 9.512 | 9.479 |
| 450 | 9.598 | 9.577 | 9.546 | 9.511 | 9.479 | 9.448 |

Tabla 5.3: Resultados de la métrica QNR para diferentes combinaciones $\beta_g$ y $\beta_u$ mediante inserción por bloques $4 \times 4$.

| $(\beta_u, \beta_g)$ | 200 | 250 | 300 | 350 | 400 | 450 |
|---|---|---|---|---|---|---|
| 200 | 42.772 | 42.484 | 42.161 | 41.799 | 41.414 | 41.016 |
| 250 | 42.489 | 42.223 | 41.911 | 41.579 | 41.215 | 40.836 |
| 300 | 42.162 | 41.914 | 41.621 | 41.308 | 40.971 | 40.620 |
| 350 | 41.784 | 41.571 | 41.306 | 41.018 | 40.701 | 40.369 |
| 400 | 41.410 | 41.211 | 40.969 | 40.705 | 40.409 | 40.099 |
| 450 | 41.029 | 40.854 | 40.620 | 40.385 | 40.112 | 39.816 |

Tabla 5.4: Resultados de valores PSNR para diferentes combinaciones $\beta_g$ y $\beta_u$ con inserción $8 \times 8$.

| $(\beta_u, \beta_g)$ | 200 | 250 | 300 | 350 | 400 | 450 |
|---|---|---|---|---|---|---|
| 200 | - | - | - | - | - | - |
| 250 | 320.156 | - | - | - | - | - |
| 300 | 360.555 | 390.512 | - | - | - | - |
| 350 | 403.113 | 430.116 | - | - | - | - |
| 400 | - | - | - | - | - | - |
| 450 | - | - | - | - | - | - |

Tabla 5.5: Energía de las combinación de $\beta_g$ y $\beta_u$ posibles mediante la inserción de bloques de $8 \times 8$.

#### *5.4.2.2. Resultados para la técnica de inserción por bloques* $8 \times 8$

De igual forma que para la técnica de inserción por bloques de $4 \times 4$ se busca la mejor combinación de los valores $\beta_g$ y $\beta_u$ mediante las 3 restricciones utilizadas, valores PSNR mayor a 41.5 dB, considerar sólo las combinaciones donde $\beta_g < \beta_u$ y por último seleccionar la combinación con la mayor energía. En la Tabla 5.4 se muestran los resultados obtenidos en la combinaciones de $\beta_g$ y $\beta_u$. Además, en la Figura 5.4 se observan las combinaciones que están dentro de la región aceptable. En la Tabla 5.5 se reportan las energías de las combinaciones con un PSNR mayor a 41.5 dB donde se eligen los valores $\beta_g = 250$ y $\beta_u = 350$. Los resultados obtenidos con respecto a la métrica QNR se presentan en la Tabla 5.6. La combinación seleccionada reporta un valor de 9.53 el cual es considerado como buena calidad.

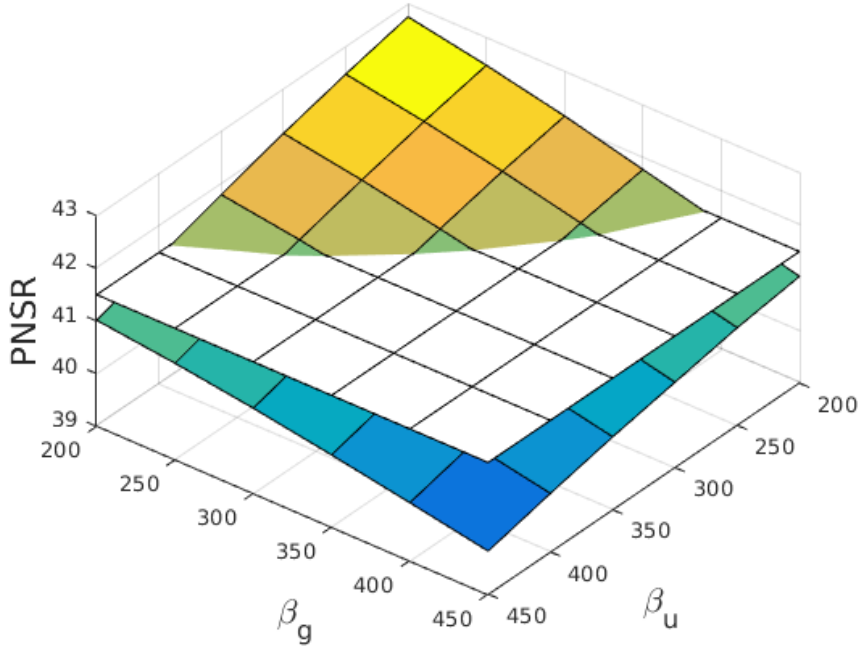


Figura 5.3: Región aceptable para valores de calidad PSNR con inserción $8 \times 8$.

| $(\beta_u, \beta_g)$ | 200 | 250 | 300 | 350 | 400 | 450 |
|---|---|---|---|---|---|---|
| 200 | 9.763 | 9.724 | 9.678 | 9.630 | 9.577 | 9.522 |
| 250 | 9.719 | 9.683 | 9.638 | 9.589 | 9.537 | 9.484 |
| 300 | 9.670 | 9.636 | 9.591 | 9.543 | 9.492 | 9.439 |
| 350 | 9.619 | 9.582 | 9.540 | 9.494 | 9.445 | 9.391 |
| 400 | 9.567 | 9.532 | 9.488 | 9.443 | 9.394 | 9.342 |
| 450 | 9.512 | 9.479 | 9.437 | 9.392 | 9.344 | 9.293 |

Tabla 5.6: Resultados de la métrica QNR para diferentes combinaciones $\beta_g$ y $\beta_u$ mediante inserción de bloques $8 \times 8$.

| $(\beta_u, \beta_g)$ | 200 | 250 | 300 | 350 | 400 | 450 |
|---|---|---|---|---|---|---|
| 200 | 43.344 | 43.026 | 42.670 | 42.284 | 41.885 | 41.474 |
| 250 | 43.028 | 42.727 | 42.395 | 42.041 | 41.663 | 41.274 |
| 300 | 42.668 | 42.397 | 42.096 | 41.769 | 41.403 | 41.032 |
| 350 | 42.285 | 42.033 | 41.761 | 41.447 | 41.112 | 40.775 |
| 400 | 41.879 | 41.655 | 41.402 | 41.114 | 40.806 | 40.488 |
| 450 | 41.467 | 41.269 | 41.033 | 40.767 | 40.487 | 40.183 |

Tabla 5.7: Resultados de valores PSNR para diferentes combinaciones $\beta_g$ y $\beta_u$ con inserción por *frame* completo.

| $(\beta_u, \beta_g)$ | 200 | 250 | 300 | 350 | 400 | 450 |
|---|---|---|---|---|---|---|
| 200 | - | - | - | - | - | - |
| 250 | 320.156 | - | - | - | - | - |
| 300 | 360.555 | 390.512 | - | - | - | - |
| 350 | 403.113 | 430.116 | 460.977 | - | - | - |
| 400 | 447.214 | 471.699 | - | - | - | - |
| 450 | - | - | - | - | - | - |

Tabla 5.8: Energía de las combinación de $\beta_g$ y $\beta_u$ posibles mediante la inserción de *frame* completo.

#### *5.4.2.3. Resultados para la técnica de inserción por frame completo.*

De forma similar la elección de la mejor combinación para $\beta_g$ y $\beta_u$ se realiza usando las restricciones utilizadas en las dos técnicas de inserción anteriores. En la Tabla 5.7 se reportan los valores PSNR obtenidos usando la técnica de inserción de *frame* completo. En la Figura 5.4 se puede apreciar las combinaciones que pertenecen a la región aceptable. La energía de la marca para las combinaciones posibles se reportan en la Tabla 5.8, con base a los resultados se seleccionan los valores $\beta_g = 250$ y $\beta_u = 400$ por tener la mayor energía. Con lo que respecta a la métrica QNR se reportan los valores obtenidos en la Tabla 5.9, en donde se observa que la combinación elegida representa una buena calidad al tener un valor de 9.83.

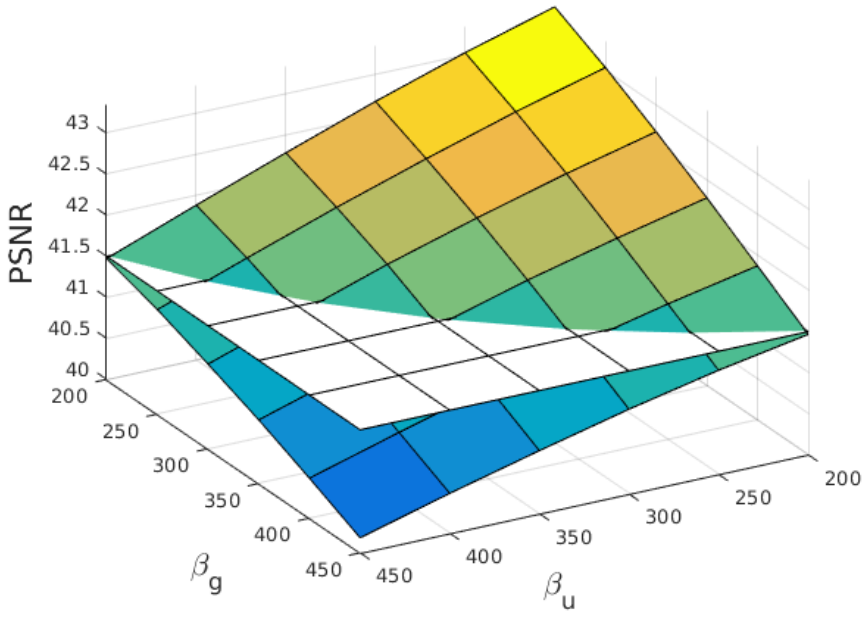


Figura 5.4: Región aceptable para valores de calidad PSNR con inserción de *frame* completo.

| $(\beta_u, \beta_g)$ | 200 | 250 | 300 | 350 | 400 | 450 |
|---|---|---|---|---|---|---|
| 200 | 9.828 | 9.831 | 9.836 | 9.839 | 9.843 | 9.844 |
| 250 | 9.828 | 9.832 | 9.835 | 9.840 | 9.842 | 9.844 |
| 300 | 9.830 | 9.833 | 9.836 | 9.841 | 9.842 | 9.844 |
| 350 | 9.829 | 9.835 | 9.837 | 9.840 | 9.842 | 9.843 |
| 400 | 9.830 | 9.835 | 9.838 | 9.839 | 9.841 | 9.844 |
| 450 | 9.832 | 9.836 | 9.838 | 9.840 | 9.841 | 9.843 |

Tabla 5.9: Resultados de la métrica QNR para diferentes combinaciones $\beta_g$ y $\beta_u$ mediante inserción por *frame* completo.

## 5.5 Pruebas de robustez

En las pruebas de robustez se evalúa la cantidad de usuarios que se detectan en un video que ha sufrido un ataque de colusión. Dichos usuarios detectados deben pertenecer al conjunto de usuarios coludidos entre sí. En las pruebas realizadas se evalúan las tres técnicas de inserción propuestas.

### 5.5.1 Escenario de prueba

El escenario de prueba consiste en una simulación de distribución legal en donde el video es distribuido a 50 usuarios. Los usuarios se dividen en 5 grupos donde cada grupo contiene 10 usuarios. A cada video se le inserta el identificador de cada uno de los usuarios. En cada prueba se generan 49 ataques en donde hay de 2 hasta 50 usuarios coludidos entre sí. Con las pruebas realizadas en la Sección 5.4 se obtuvieron combinaciones de $\beta_g$ y $\beta_u$ diferentes para cada técnica de inserción. Sin embargo, se decide utilizar los valores $\beta_g$ y $\beta_u$ iguales para las 3 técnicas de inserción en la evaluación de la robustez para no ser un criterio de desigualdad. Con este ajuste se tiene una misma energía de marca en los 3 casos y así el criterio de evaluación de las técnicas radica sólo en el número de usuarios coludidos entre sí detectados. Debido a lo anterior, el identificador es generado con un peso $\beta_g = 250$ y $\beta_u = 350$, ya que estos valores cumplen las restricciones para las 3 técnicas de inserción, así como una longitud fija de $l_g = 1024$ y $l_u = 1024$. Con cada técnica propuesta se evalúan cinco tipos de ataques: promedio, mínimo, máximo, mediana y minmax. Dicho escenario de prueba se replica para cada uno de los 30 videos disponibles por lo que se reporta la mediana de las detecciones de cada ataque. En la Figura 5.5 se muestra el escenario de la prueba.

### 5.5.2 Sintonización de valores $p_e$ y $p_{ec}$

Para sintonizar los valores de probabilidad de error en la detección de identificador ($p_e$) y la probabilidad de error para el sistema de conteo ($P_{ec}$) se realizan pruebas preliminares con un ataque

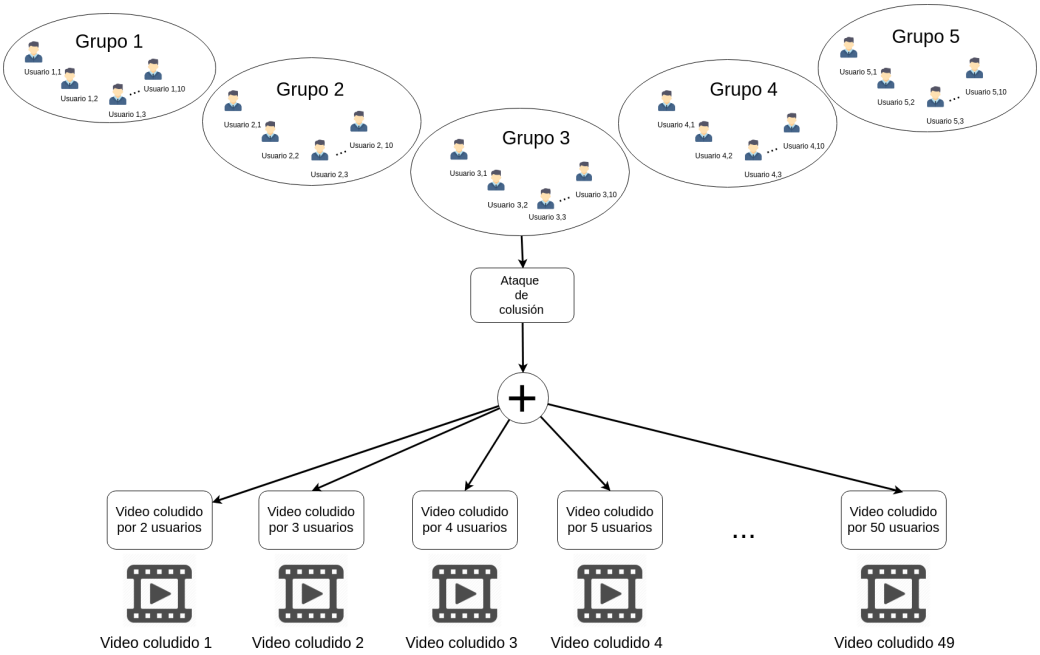


Figura 5.5: Escenario de pruebas. Los 50 usuarios son divididos en 5 grupos con 10 usuarios cada grupo. Se generan un total de 49 ataques por video en donde hay desde 2 hasta 50 usuarios coludidos entre sí.

de colusión promedio ya que es el más reportando en la literatura debido a que es el que usualmente se utiliza por los usuarios deshonestos. En las pruebas se evalúan los valores de $p_e = 10^{-5}$, $10^{-6}$ y $10^{-7}$, en cada uno de los casos la segunda probabilidad de error se evalúa $p_{ec} = 10^{-6}$, $10^{-7}$, $10^{-8}$, $10^{-9}$, $10^{-10}$, $10^{-11}$ y $10^{-12}$. A partir de esta sintonización se harán las pruebas para el resto de los ataques. La sintonización se realiza para cada una de las técnicas de inserción.

#### *5.5.2.1. Resultados para la técnica de inserción por bloques* $4 \times 4$

Los resultados de detección son reportados en las Figuras 5.6, 5.7 y 5.8, para una probabilidad de error de $p_e = 10^{-5}$, $p_e = 10^{-6}$ y $p_e = 10^{-7}$ respectivamente.

Para el caso de una probabilidad $p_e = 10^{-5}$ se logra una detección de hasta 25 usuarios coludidos entre sí, sin embargo, existe una alta detección de falsos dispositivos. Con una probabilidad $p_e = 10^{-6}$ la detección es de hasta 23 usuarios con aún detección de falsos positivos, Finalmente, para el caso de una probabilidad $p_e = 10^{-7}$ se logra un detección de 20 usuarios coludidos entre sí con algunos

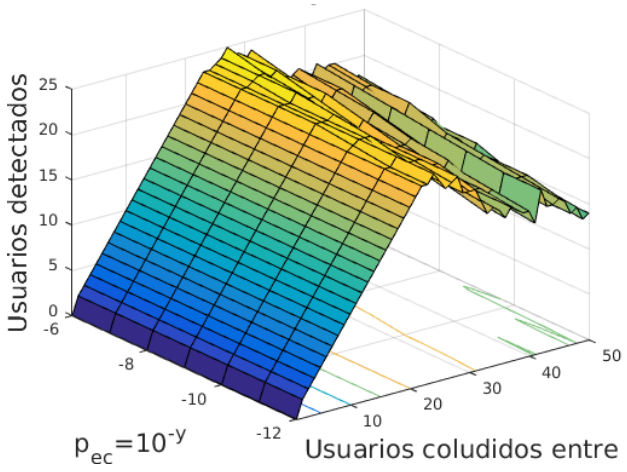


Figura 5.6: Resultados de la detección de usuarios con una probabilidad $p_e = 10^{-5}$ para inserción por bloques de $4 \times 4$.

pocos falsos positivos, pero en combinación con una probabilidad $p_{ec} = 10^{-7}$ la detección de falsos positivos desaparecen.

Tomando en cuenta los resultados anteriores se decide una detección baja con la seguridad de que no exista falsos positivos y con esto se evita acusar a usuarios inocentes. De esta forma los valores establecidos son: $p_e = 10^{-7}$ y $p_{ec} = 10^{-7}$. Con estos valores se tuvo una detección menor a la obtenida con $p_e = 10^{-5}$ y $p_e = 10^{-6}$, algo esperado dado que a menor probabilidad el proceso de detección es más discriminante.

#### *5.5.2.2. Resultados para la técnica de inserción por bloques* $8 \times 8$

Los resultados obtenidos para la inserción por bloques de $8 \times 8$ son reportados en las Figuras 5.9, 5.10 y 5.11, para una probabilidad de error de $p_e = 10^{-5}$, $p_e = 10^{-6}$ y $p_e = 10^{-7}$ respectivamente. De manera similar a la técnica de inserción $4$, para los valores $p_e = 10^{-5}$ y $p_e = 10^{-6}$ existe detección de falsos positivos con una detección máxima de 23 y 21 respectivamente. Los valores de probabilidad de error $p_e = 10^{-7}$ y $p_{ec} = 10^{-8}$ son seleccionados ya que no se detecta falsos positivos con una detección máxima de 20 usuarios coludidos entre sí.

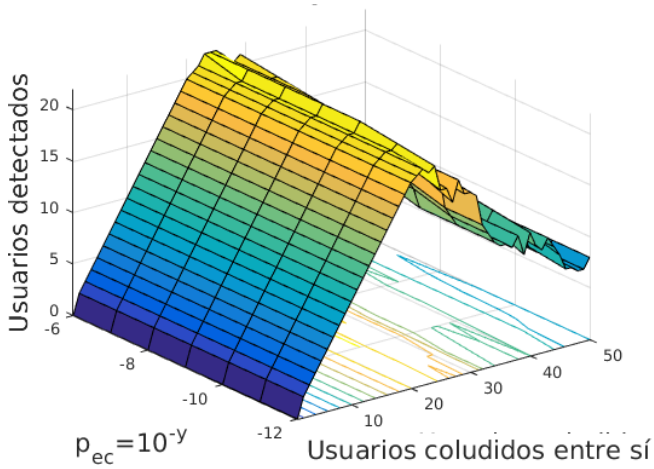


Figura 5.7: Resultados de la detección de usuarios con una probabilidad $p_e = 10^{-6}$ para inserción por bloques de $4 \times 4$.

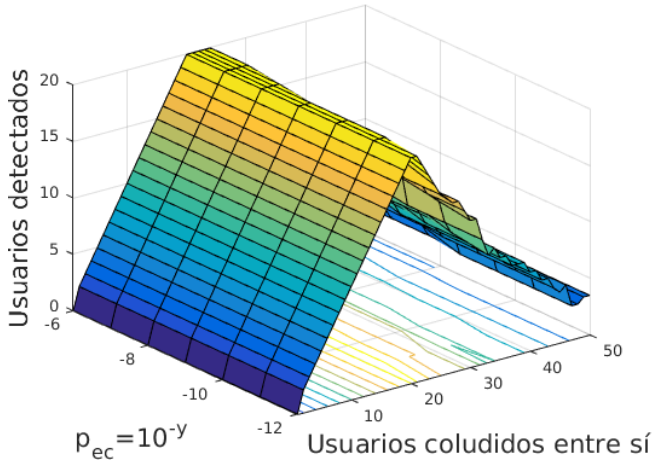


Figura 5.8: Resultados de la detección de usuarios con una probabilidad $p_e = 10^{-7}$ para inserción por bloques de $4 \times 4$.

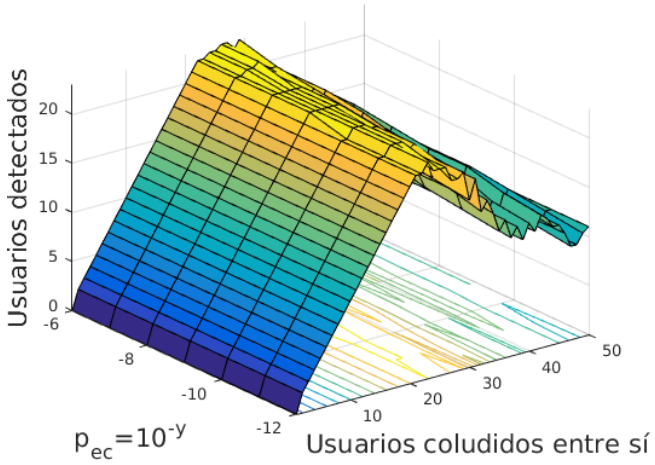


Figura 5.9: Resultados de la detección de usuarios con una probabilidad $p_e = 10^{-5}$ para inserción por bloques de $8 \times 8$.

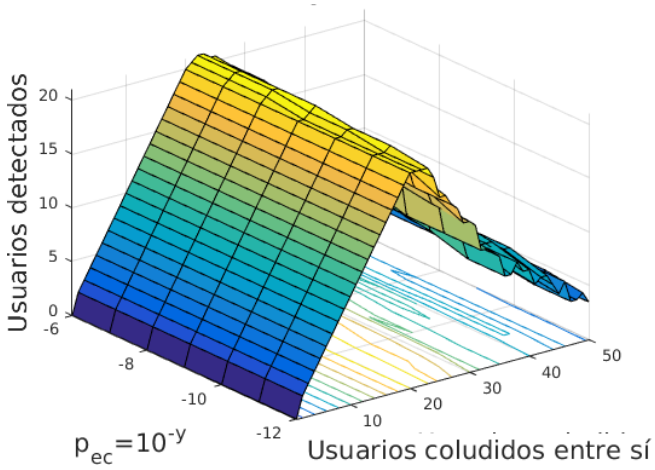


Figura 5.10: Resultados de la detección de usuarios con una probabilidad $p_e = 10^{-6}$ para inserción por bloques de $8 \times 8$.

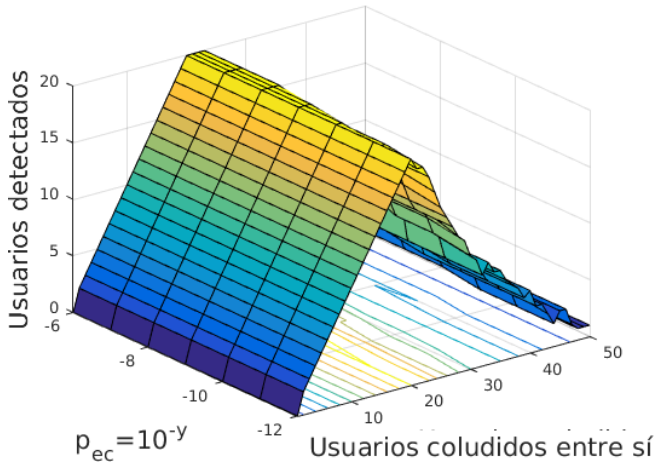


Figura 5.11: Resultados de la detección de usuarios con una probabilidad $p_e = 10^{-7}$ para inserción por bloques de $8 \times 8$.

#### *5.5.2.3. Resultados para la técnica de inserción por frame completo*

La detección de usuarios coludidos entre sí con el uso de la técnica de inserción por *frame* completo son reportados en las Figuras 5.12, 5.13 y 5.14, para una probabilidad de error de $p_e = 10^{-5}$, $p_e = 10^{-6}$ y $p_e = 10^{-7}$ respectivamente. De igual forma que las pruebas de detección para las otras dos técnicas de inserción, con los valores de $p_e = 10^{-5}$ y $p_e = 10^{-6}$ se presentan detecciones de falso positivo con una detección máxima de 25 usuarios coludidos entre sí para $p_e = 10^{-5}$ y 23 para $p_e = 10^{-6}$. Con los valores $p_e = 10^{-7}$ y $p_{ec} = 10^{-6}$ no se detectan falsos positivos y se tiene una detección máxima de 22 usuarios coludidos entre sí.

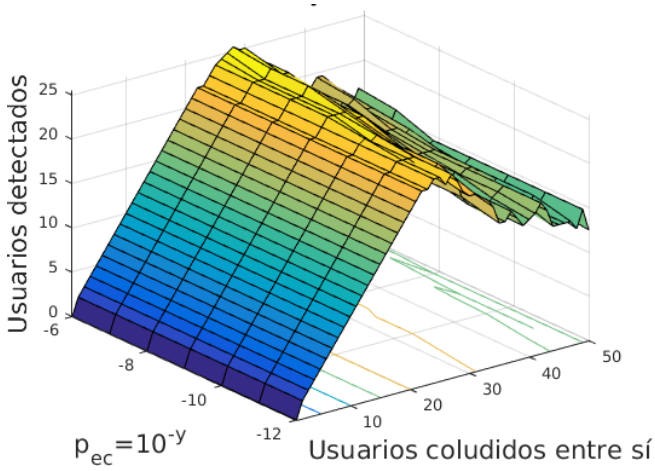


Figura 5.12: Resultados de la detección de usuarios con una probabilidad $p_e = 10^{-5}$ para inserción por *frame* completo.

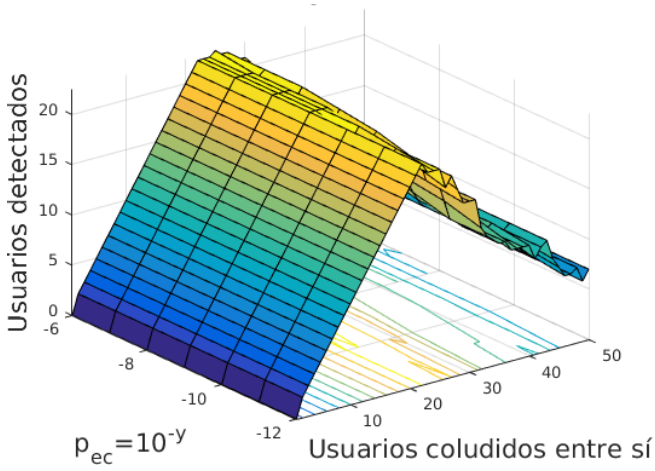


Figura 5.13: Resultados de la detección de usuarios con una probabilidad $p_e = 10^{-6}$ para inserción por *frame* completo.

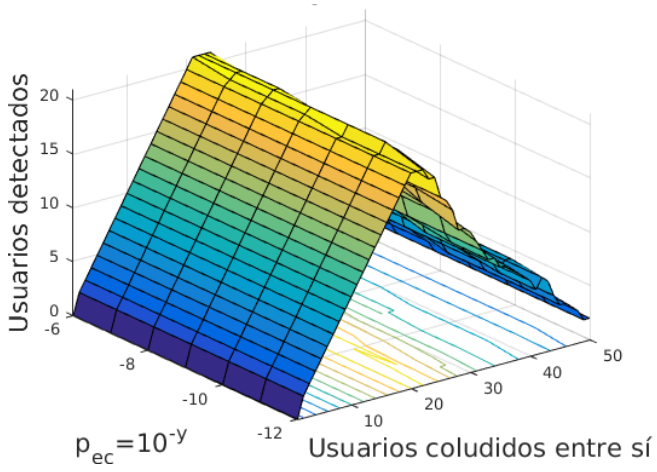


Figura 5.14: Resultados de la detección de usuarios con una probabilidad $p_e = 10^{-7}$ para inserción por *frame* completo.

### 5.5.3 Prueba de tiempo mínimo para máxima detección

Para determinar el tiempo mínimo de video necesario para realizar la máxima detección de usuarios coludidos entre sí se realiza una prueba donde se usan 5 segundos de video (125 *frames* con una tasa de 25 fps). Se utiliza la técnica de inserción por *frame* completo dado que con ésta, en pruebas preliminares con los diferentes ataques, se obtuvieron mejores resultados lo cual se puede corroborar con los resultados de las Secciones 5.5.4, 5.5.5 y 5.5.6. Se utiliza el ataque promedio ya que, como se mencionó en las pruebas de sintonización de $\beta_g$ y $\beta_u$, es el más utilizado en la literatura. Con esta prueba se realiza la detección de usuarios cada segundo para determinar a partir de qué segundo el número de usuarios detectados se mantiene constante.

En la Figura 5.15 se muestra de manera conjunta los resultados obtenidos en la detección para cada segundo. Por otra parte, en la Tablas 5.10 y 5.11 se reportan de manera específica el número de usuarios detectados en cada segundo. Con base a que la detección de usuarios a partir de 2 segundos es, si bien no constante, prácticamente igual por lo que se puede definir que el tiempo mínimo para una máxima detección es de 2 segundos.

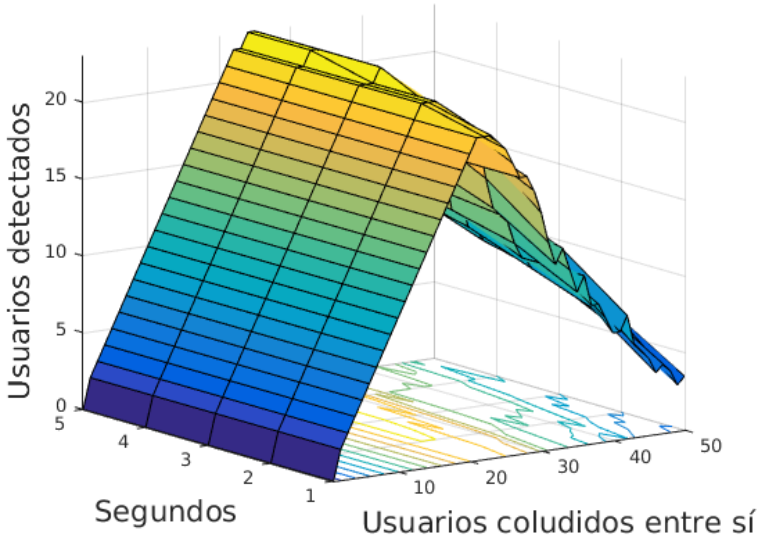


Figura 5.15: Resultados de detección de usuarios coludidos entre sí para diferentes segundos de video.

| Número de Coludidos | Tiempo de video (seg) | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| 2 | 2 | 2 | 2 | 2 | 2 |
| 3 | 3 | 3 | 3 | 3 | 3 |
| 4 | 4 | 4 | 4 | 4 | 4 |
| 5 | 5 | 5 | 5 | 5 | 5 |
| 6 | 6 | 6 | 6 | 6 | 6 |
| 7 | 7 | 7 | 7 | 7 | 7 |
| 8 | 8 | 8 | 8 | 8 | 8 |
| 9 | 9 | 9 | 9 | 9 | 9 |
| 10 | 10 | 10 | 10 | 10 | 10 |
| 11 | 11 | 11 | 11 | 11 | 11 |
| 12 | 12 | 12 | 12 | 12 | 12 |
| 13 | 13 | 13 | 13 | 13 | 13 |
| 14 | 14 | 14 | 14 | 14 | 14 |
| 15 | 15 | 15 | 15 | 15 | 15 |
| 16 | 16 | 16 | 16 | 16 | 16 |
| 17 | 17 | 17 | 17 | 17 | 17 |
| 18 | 18 | 18 | 18 | 18 | 18 |
| 19 | 19 | 19 | 19 | 19 | 19 |
| 20 | 20 | 20 | 20 | 20 | 20 |
| 21 | 21 | 21 | 21 | 21 | 21 |
| 22 | 21 | 22 | 22 | 22 | 22 |
| 23 | 21 | 22 | 22 | 22 | 22 |
| 24 | 21 | 22 | 22 | 23 | 23 |
| 25 | 20 | 21 | 23 | 23 | 23 |

Tabla 5.10: Resultados de detección de usuarios coludidos entre sí para diferente tiempo de video (2 a 25 coludidos).

| Número de Coludidos | Tiempo de video (seg) | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| 26 | 19 | 21 | 21 | 22 | 23 |
| 27 | 18 | 18 | 21 | 22 | 22 |
| 28 | 18 | 20 | 21 | 22 | 22 |
| 29 | 17 | 19 | 20 | 21 | 21 |
| 30 | 16 | 19 | 20 | 22 | 22 |
| 31 | 14 | 17 | 19 | 21 | 21 |
| 32 | 12 | 15 | 16 | 18 | 18 |
| 33 | 13 | 14 | 15 | 16 | 17 |
| 34 | 12 | 14 | 16 | 17 | 17 |
| 35 | 10 | 12 | 13 | 14 | 15 |
| 36 | 11 | 11 | 13 | 14 | 15 |
| 37 | 9 | 12 | 13 | 14 | 15 |
| 38 | 9 | 12 | 13 | 14 | 15 |
| 39 | 8 | 12 | 13 | 14 | 16 |
| 40 | 7 | 12 | 13 | 14 | 15 |
| 41 | 7 | 10 | 13 | 14 | 16 |
| 42 | 8 | 11 | 13 | 13 | 14 |
| 43 | 6 | 10 | 12 | 13 | 15 |
| 44 | 6 | 9 | 11 | 13 | 15 |
| 45 | 5 | 8 | 10 | 12 | 13 |
| 46 | 4 | 7 | 9 | 10 | 11 |
| 47 | 5 | 7 | 9 | 10 | 13 |
| 48 | 4 | 6 | 8 | 9 | 11 |
| 49 | 3 | 7 | 9 | 11 | 12 |
| 50 | 4 | 6 | 8 | 10 | 11 |

Tabla 5.11: Resultados de detección de usuarios coludidos entre sí para diferente tiempo de video (26 a 50 coludidos).

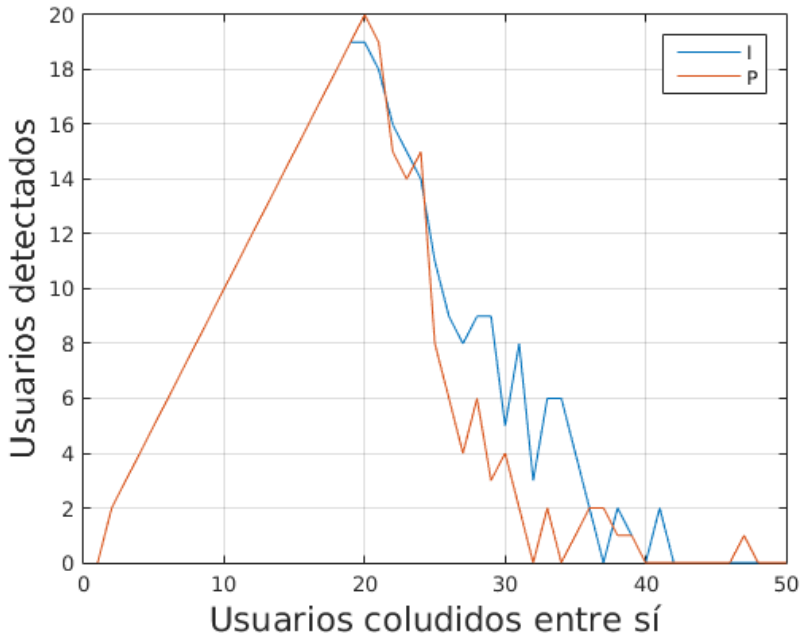


Figura 5.16: Detección de usuarios coludidos entre sí por tipo de *frame* para un perfil *Baseline*.

### 5.5.4 Resultados de robustez usando bloques de $4 \times 4$

Como se comentó, en un perfil *Baseline* se utilizan dos tipos de *frame*, *frame I* y *frame P*. Se realiza una prueba preliminar contemplando solo *frames* tipo *I* y posteriormente otra donde se contempla los *frames* tipo *P*. En la Figura 5.16 se muestran los resultados obtenidos para ambas pruebas, si bien se obtiene una mayor detección con *frame I* el uso de ambos *frame* aporta una mayor votación para el sistema de conteo de votos (descrito en la Sección 4.2.4).

Los resultado obtenidos para los diferentes ataques se presentan en la Figura 5.17 donde se observa que el ataque en el cual se detectan más usuarios es el ataque promedio, mientras que los ataques más efectivos son los ataques mínimo, máximo y minmax.

En la Tabla 5.12 se muestra una comparación entre la detección obtenida usando la técnica de inserción de bloques de $4 \times 4$ y la detección reportada en [42, 46]. Como se puede observar los resultados obtenidos para el ataque de colusión promedio son competitivos a los reportados en los trabajos [46] y en [42], sin embargo, para el resto de los ataques se presentan algunas detecciones

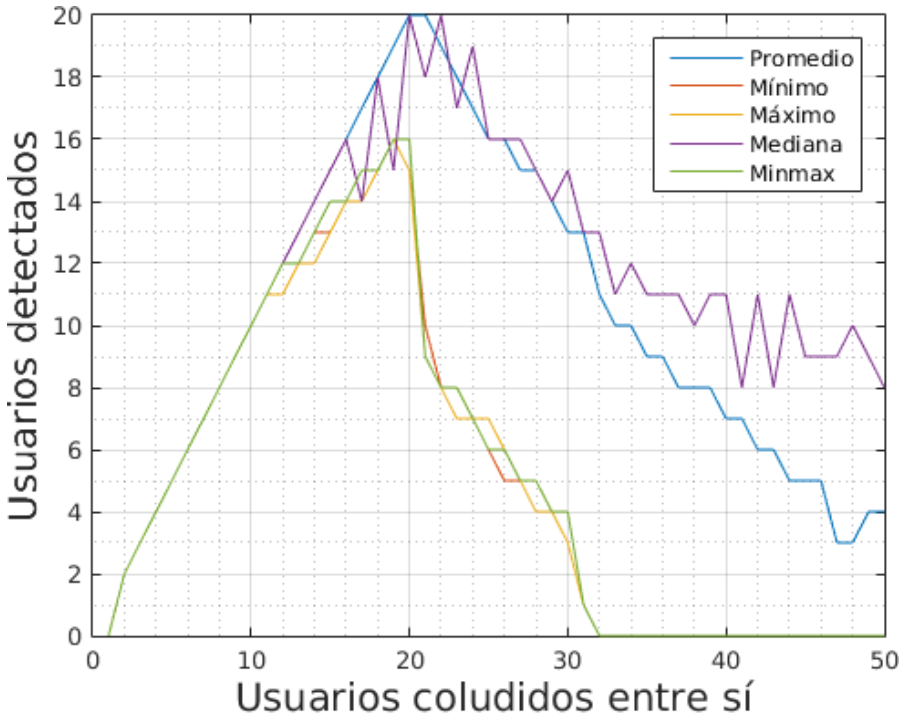


Figura 5.17: Detección de usuarios coludidos entre sí usando la técnica de inserción de bloques $4 \times 4$ para diferentes ataques de colusión.

por debajo en los reportados en dichos trabajos.

### 5.5.5 Resultados de robustez usando bloques de $8 \times 8$

Como se ha mencionado para el caso de un perfil *High* se presentan 3 tipos de *frame*, *frame I*, *frame P* y *frame B*. Para probar el efecto de elegir algún tipo de *frame* se realiza una prueba preliminar en la que se hace la detección utilizando un solo tipo de *frame*. Los resultados se muestran en la Figura 5.18 donde se observa que con los *frame I* se logra una mayor detección seguido por los *frame P* y posteriormente los *frame B*. Dado que cada tipo de *frame* aportan detecciones correctas se decide utilizar los tres tipos ya que aportarán información al sistema de conteo de votos.

Los resultado obtenidos para los diferentes ataques se presentan en la Figura 5.19 donde se puede observar que el ataque en el cual se detectan más usuarios es el ataque promedio, mientras que los ataques más efectivos son los ataques mínimo y máximo.

| Propuesta | Número de Coludidos | Número de usuarios detectados por tipo de colusión | | | | |
|---|---|---|---|---|---|---|
| | | Avg | Min | Max | Med | MinMax |
| [46] | 11 | 11 | 10 | 10 | 10 | 10 |
| [42] | | 10 | 10 | 10 | 10 | 10 |
| Bloque $4 \times 4$ | | 11 | 11 | 11 | 11 | 11 |
| [46] | 14 | 14 | 13 | 13 | 13 | 13 |
| [42] | | 14 | 14 | 14 | 13 | 14 |
| Bloque $4 \times 4$ | | 14 | 13 | 12 | 14 | 13 |
| [46] | 17 | 16 | 15 | 14 | 14 | 14 |
| [42] | | 15 | 16 | 16 | 16 | 15 |
| Bloque $4 \times 4$ | | 17 | 14 | 14 | 14 | 15 |
| [46] | 20 | 18 | 16 | 16 | 16 | 17 |
| [42] | | 17 | 17 | 17 | 16 | 17 |
| Bloque $4 \times 4$ | | 20 | 15 | 15 | 20 | 16 |

Tabla 5.12: Comparación del número de usuarios coludidos entre sí detectados usando la técnica de inserción por bloques de $4 \times 4$ con los reportados en [42, 46].

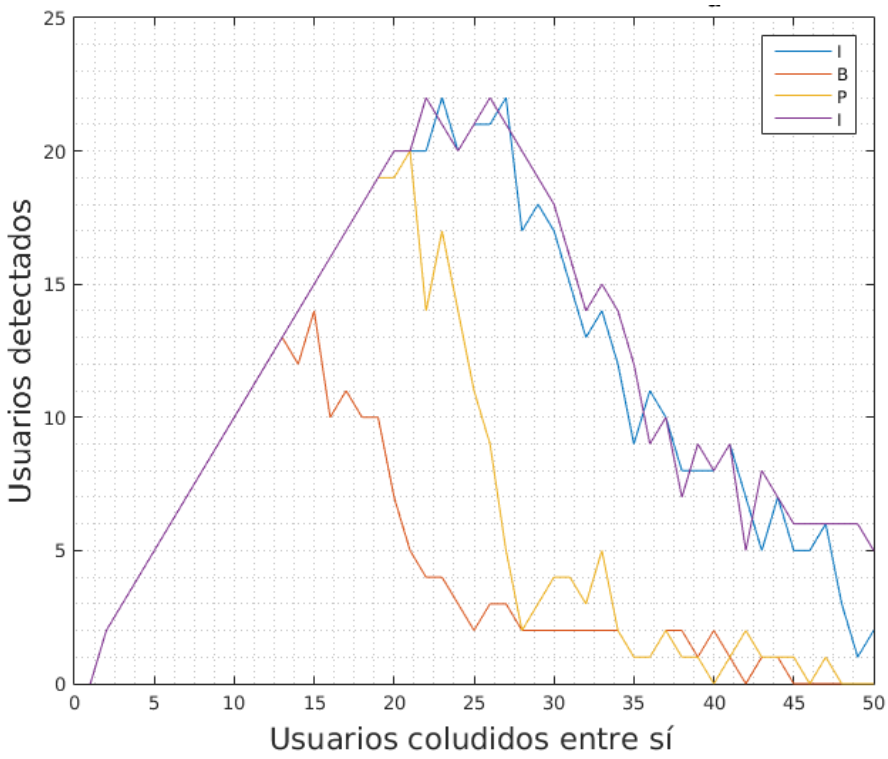


Figura 5.18: Detección de usuarios coludidos entre sí por tipo de *frame* para un perfil *High*.

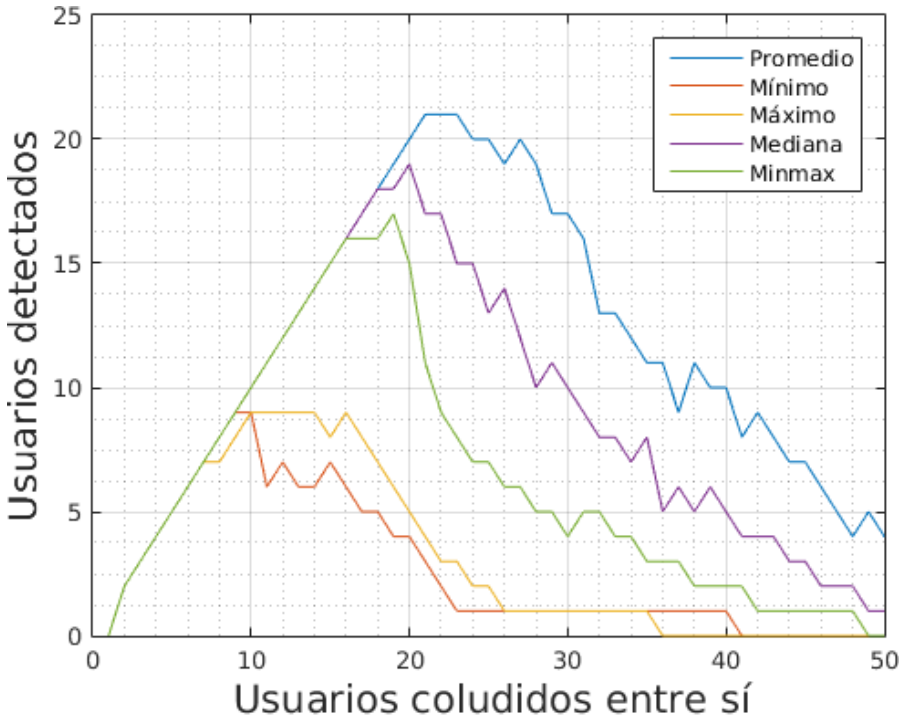


Figura 5.19: Detección de usuarios coludidos entre sí usando la técnica de inserción de bloques $8 \times 8$ para diferentes ataques de colusión.

En la Tabla 5.13 se muestra una comparación entre la detección de usuarios que se obtiene usando la técnica de inserción de bloques de $8 \times 8$ y la detección reportada en [42, 46]. Como se puede observar los resultados obtenidos para el ataque de colusión promedio, mediana y minmax son competitivos a los reportados en los trabajos [46] y en [42], sin embargo, para los ataques mínimo y máximo se presentan un número de detecciones por debajo a los reportados.

### 5.5.6 Resultados de robustez usando el *frame* completo

Para el caso de la técnica de inserción mediante *frame* completo se evalúa el rendimiento para ambos perfiles de compresión. Los resultado obtenidos para los diferentes ataques bajo el perfil *Baseline* se presentan en la Figura 5.20 donde se percibe que el ataque en el que se detectan más usuarios es el ataque promedio, mientras que el ataque más efectivo son los ataques mínimo y máximo.

| Propuesta | Número de Coludidos | Número de usuarios detectados por tipo de colusión | | | | |
|---|---|---|---|---|---|---|
| | | Avg | Min | Max | Med | MinMax |
| [46] | 11 | 11 | 10 | 10 | 10 | 10 |
| [42] | | 10 | 10 | 10 | 10 | 10 |
| Bloque $8 \times 8$ | | 11 | 6 | 9 | 11 | 11 |
| [46] | 14 | 14 | 13 | 13 | 13 | 13 |
| [42] | | 14 | 14 | 14 | 13 | 14 |
| Bloque $8 \times 8$ | | 14 | 6 | 9 | 11 | 11 |
| [46] | 17 | 16 | 15 | 14 | 14 | 14 |
| [42] | | 15 | 16 | 16 | 16 | 15 |
| Bloque $8 \times 8$ | | 17 | 5 | 8 | 17 | 16 |
| [46] | 20 | 18 | 16 | 16 | 16 | 17 |
| [42] | | 17 | 17 | 17 | 16 | 17 |
| Bloque $8 \times 8$ | | 20 | 4 | 5 | 19 | 20 |

Tabla 5.13: Comparación del número de usuarios coludidos entre sí detectados usando la técnica de inserción por bloques de $8 \times 8$ con los reportados en [42, 46].

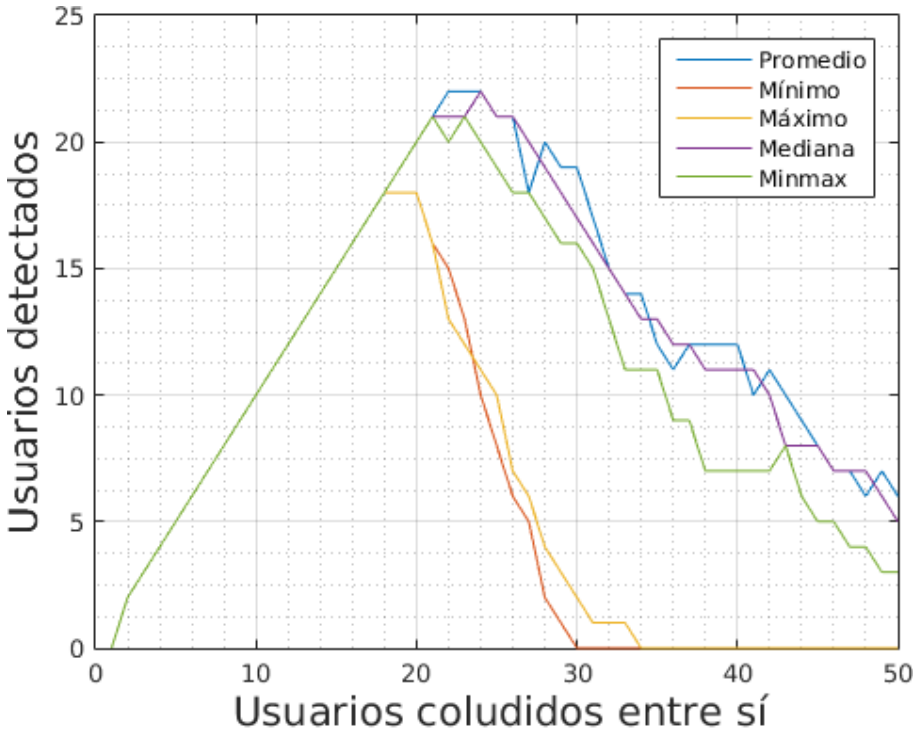


Figura 5.20: Detección de usuarios coludidos entre sí usando la técnica de inserción mediante *frame* completo para diferentes ataques de colusión bajo el perfil *Baseline*.

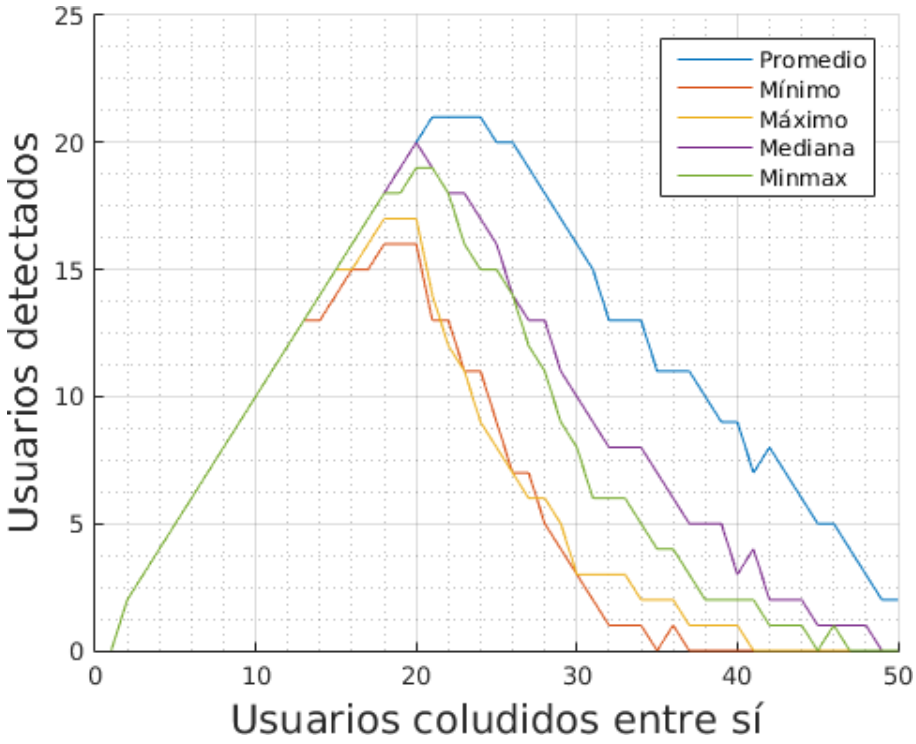


Figura 5.21: Detección de usuarios coludidos entre sí usando la técnica de inserción mediante *frame* completo para diferentes ataques de colusión bajo el perfil *High*.

En la Figura 5.21 se muestran los resultados de la detección de usuarios bajo el perfil *High*, se observa que el ataque en el que se detectan más usuarios es el ataque promedio, mientras que el ataque más efectivo son los ataques mínimo y máximo.

En la Tabla 5.14 se muestra una comparación entre la detección de usuarios que se obtiene usando la técnica de inserción de *frame* completo bajo los dos perfiles de compresión y la detección reportada en [42, 46]. Como se puede observar los resultados obtenidos para todos los ataques son superiores a los reportados en los trabajos [46] y en [42].

## 5.6 Pruebas de capacidad

Con base a las pruebas realizadas en las Secciones 5.4 y 5.5 se puede determinar que la longitud del identificador utilizado en el método propuesto es inferior a la longitud utilizada en [42, 46], así

| Propuesta | Número de Coludidos | Número de usuarios detectados por tipo de colusión | | | | |
|---|---|---|---|---|---|---|
| | | Avg | Min | Max | Med | MinMax |
| [46] | 11 | 11 | 10 | 10 | 10 | 10 |
| [42] | | 10 | 10 | 10 | 10 | 10 |
| Propuesto (*Baseline*) | | 11 | 11 | 11 | 11 | 11 |
| Propuesto (*High*) | | 11 | 11 | 11 | 11 | 11 |
| [46] | 14 | 14 | 13 | 13 | 13 | 13 |
| [42] | | 14 | 14 | 14 | 13 | 14 |
| Propuesto (*Baseline*) | | 14 | 14 | 14 | 14 | 14 |
| Propuesto (*High*) | | 14 | 14 | 14 | 14 | 14 |
| [46] | 17 | 16 | 15 | 14 | 14 | 14 |
| [42] | | 15 | 16 | 16 | 16 | 15 |
| Propuesto (*Baseline*) | | 17 | 17 | 17 | 17 | 17 |
| Propuesto (*High*) | | 17 | 16 | 17 | 17 | 17 |
| [46] | 20 | 18 | 16 | 16 | 16 | 17 |
| [42] | | 17 | 17 | 17 | 16 | 17 |
| Propuesto (*Baseline*) | | 20 | 18 | 18 | 20 | 20 |
| Propuesto (*High*) | | 20 | 17 | 18 | 20 | 20 |

Tabla 5.14: Comparación del número de usuarios coludidos entre sí detectados usando la técnica de inserción por *frame* completo con las reportadas en [42, 46].

también, se aprecia que se tiene un número igual o mayor de usuarios detectados. En el caso de las prueba de ataque de promedio con una técnica de inserción con *frame* completo se obtiene una detección máxima de 22 usuarios coludidos entre sí. Con el enfoque propuesto la longitud utilizada ($l_n$) se calcula como:

$$l_n = (l_g + l_u) \times nframe \tag{5.2}$$

donde:

$l_g$ es la longitud del identificador correspondiente al grupo, $l_u$ es la longitud correspondiente al usuario y $nframe$ es el número de *frames* necesarios para la detección. De tal modo que:

$$l_n = (1024 + 1024) \times 50 = 1.024 \times 10^5 \tag{5.3}$$

Por otra parte, si se utilizara códigos de Tardos cuya longitud está dada por:

$$l_n = 100c^2 \times \ln(1/\varepsilon_1) \tag{5.4}$$

donde $c$ es el número de usuarios coludidos entre sí y $\varepsilon$ la probabilidad de error. Bajo los mismos parámetros de las pruebas realizadas ( $c = 22$ y $\varepsilon = 10^{-7}$) la longitud necesaria sería:

$$l_n = 100(22)^2 \times \ln(1/10^{-7}) = 7.80 \times 10^5 \tag{5.5}$$

En la Figura 5.22 se muestra gráficamente la diferencia entre las dos longitudes para una detección de 2 hasta 22 usuarios coludidos entre sí. Se observa que utilizando códigos de Tardos la longitud crece cuadráticamente mientras que la longitud utilizada en el método propuesto se mantiene constante.

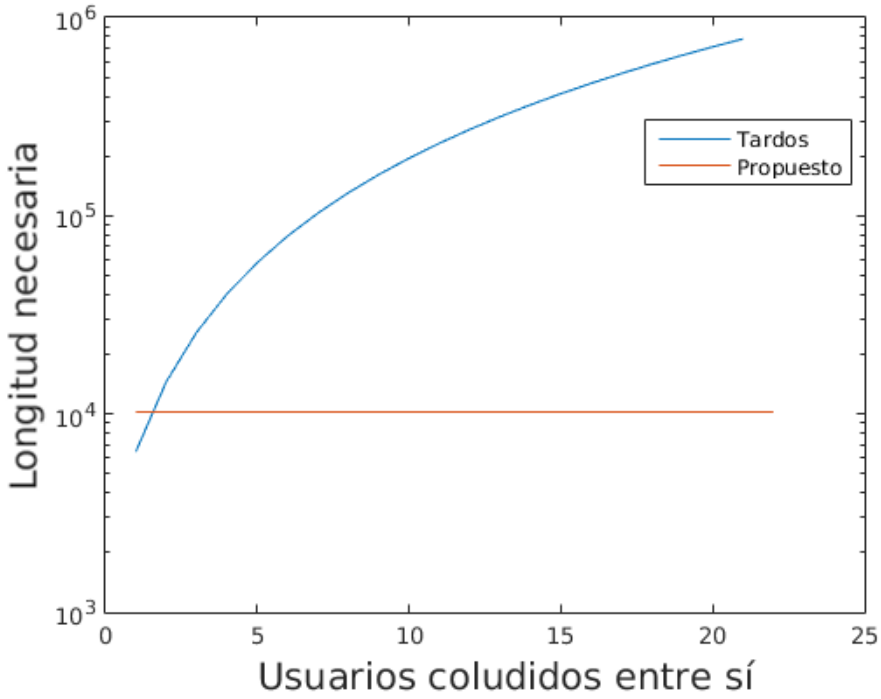


Figura 5.22: Comparativa de longitud utilizando códigos de Tardos y el propuesto.

## 5.7 Discusión

En este capítulo se reportaron los resultados obtenidos en las experimentaciones realizadas al método de rastreo de copias de video piratas propuesto. Primero se realiza una experimentación de calidad visual evaluando las métricas PSNR y QNR, se determina la energía de la marca a insertar mediante la combinación de diferentes valores de $\beta_g$ y $\beta_u$. Son evaluadas las 3 técnicas de inserción definiendo como mejor combinación los valores: para inserción por bloques de $4 \times 4$, $\beta_g = 250$ y $\beta_u = 450$, para la inserción por bloques de $8 \times 8$, $\beta_g = 250$ y $\beta_u = 350$, finalmente, para la inserción por *frame* completo $\beta_g = 250$ y $\beta_u = 400$. Con los resultados de calidad visual obtenidos para cada una de las técnicas de inserción se puede determinar que la técnica de inserción con bloques de $8 \times 8$ obtiene menores valores de PSNR, lo que demuestra que esta técnica de inserción genera mayor perdida de calidad visual con respecto al uso de las otras dos técnicas de inserción. Aunque cabe destacar que en promedio la diferencia entre las técnicas de inserción por *frame* completo y por

bloques de $4 \times 4$ es de apenas 0.45 dB y 0.70 dB respectivamente, lo cual indica que la diferencia de calidad entre el uso de una técnica de inserción y otra es muy pequeña. Por el contrario la inserción por bloques de $4 \times 4$ proporciona mayor calidad bajo la métrica PSNR debido a que, como se mencionó, la métrica PSNR calcula el MSE de la imagen editada con respecto a la original y al realizar la inserción con bloques sólo se modifica el valor de los píxeles de los bloques en los que hubo inserción, de tal modo que se modifican menos píxeles al trabajar con bloques $4 \times 4$.

Por otro lado, referente a la métrica QNR la inserción por *frame* completo arroja los mejores resultados, esto podría deberse al efecto de bloque generado por trabajar con la inserción por bloques, dado que al utilizar el *frame* completo no existe dicho efecto se obtiene una mejor calidad perceptual.

Se realizaron experimentaciones para sintonizar los valores de $p_e$ y $p_{ec}$ para cada uno de las técnicas de inserción, donde se determinó que para las tres técnicas el mejor valor para la probabilidad de error de detección de identificador es $p_e = 10^{-7}$ mientras que la probabilidad de error para el vector de votos $p_{ec}$ varía según la técnica de inserción. Para la inserción por bloques $4 \times 4$, $p_{ec} = 10^{-7}$, para la inserción por bloques $8 \times 8$, $p_{ec} = 10^{-8}$ y para la inserción por *frame* completo $p_{ec} = 10^{-6}$.

En otra de las experimentaciones se determinó el tiempo necesario del video para tener la máxima detección de usuarios coludidos entre sí, en esta experimentación se define 2 segundos como tiempo mínimo para dicha detección. Si bien en los resultados se observa que con más tiempo de video se obtiene un aumento de detección máxima de usuarios, este aumento es de apenas un usuario más y el costo computacional para procesar la detección aumenta considerablemente, por esta razón se decide fijar el tiempo necesario a 2 segundos.

Dado que se detectaron identificadores de manera correcta en los videos después de la codificación y decodificación podemos asegurar que los identificadores propuestos soporta la pérdida de información derivada de la compresión del estándar H.264/AVC, ello se debe a que la inserción

se realiza en frecuencias bajas y medias las cuales no son afectadas por dicha perdida.

Otras de las experimentaciones realizadas son para determinar la mejor técnica de inserción de las 3 propuestas. En esta experimentación se ponen a prueba las 3 técnicas de inserción bajo los 5 ataques de colusión descritos anteriormente. Para el ataque de colusión promedio las tres técnicas resultan competitivas ya que igualan y mejoran el número de usuarios detectados en [42, 46]. Sin embargo, para el resto de los ataques las técnicas de inserción por bloque de $4\times4$ y $8\times8$ reporta detecciones por debajo a los reportados en los trabajos mencionados. Con base a los resultados obtenidos la técnica de inserción por *frame* completo resultó ser mejor al obtener un mayor número de usuarios coludidos entre sí detectados para todos los distintos ataques de colusión. La baja eficiencia con las técnicas de inserción por bloque ante los ataques de colusión mínimo, máximo y, en el caso de inserción por bloque $4 \times 4$, minmax se atribuye principalmente a que mientras el número de píxeles a transformar sea menor la resolución frecuencial la resolución obtenida con la DCT es baja, debido a ello al modificar un coeficiente frecuencial, por la inserción de uno varios componentes del identificador, al regresar al dominio espacial existe un cambio en los valor de los píxeles pertenecientes al bloque. Dicho cambio puede ser o no agresivo, eso depende del valor del componente del identificador, sin embargo, no se tiene control sobre dicho valor ya que depende en gran medida a la señal de ruido pseudoaleatorio. Esto genera que al utilizar los ataques agresivos de mínimo o máximo se tomen los valores extremos de cada uno de los videos lo que provoca que al generar el identificador a partir de los componentes DCT existan grandes cambios y por ende se obtenga una menor detección de usuarios.

En cambio, ese efecto no es tan marcado en la técnica de inserción por *frame* completo debido a que se tiene un gran número de píxeles lo que provoca que en el dominio frecuencial exista una alta resolución. Esto ocasiona que el cambio, aunque sea muy abrupto, de los componentes DCT generado por la inserción del identificador se disperse a lo largo de todos los píxeles del *frame*, esto genera que no haya un gran cambio a los valores de los píxeles lo que se ve reflejado en una mayor

detección de usuarios.

Finalmente, se corrobora que la longitud del identificador en el método propuesto es menor al utilizado en [42, 46], lo que se ve reflejado en una mayor capacidad en relación al número de usuarios coludidos entre sí detectados y menor longitud de identificador. El uso de identificadores de menor longitud también se puede ver reflejado en el número de *frames* necesarios para la detección, en el método propuesto se necesitan, como se mencionó anteriormente, 50 *frames* mientras que en [42, 46] se necesita aproximadamente 9000 *frames*.

# 6

# Conclusiones y trabajo a futuro

## 6.1 Conclusiones

En este trabajo se presentó un método de rastreo de copias de video en formato H.264/AVC resistente a ataques de colusión. Con base a los resultados obtenidos se puede concluir que una método de *fingerprinting* que hace uso de un identificador basado en una secuencia de espectro disperso y una técnica de marcado de agua de espectro disperso, aplicado a videos comprimidos con el estándar H.264/AVC representa una buena alternativa para hacer frente a los ataques de colusión ya que se logra un número de detección competitivo con los que se reportan en el estado del arte. La inserción con el método propuesto no genera perdida de calidad visual dado que se obtienen valores aceptables bajo las métricas de PSNR y QNR. Así mismo, se puede concluir que el tiempo necesario de video para realizar una máxima detección es de 2 segundos. La utilización de un esquema de votación permite que, a diferencia de utilizar un sólo umbral, la detección de usuarios coludidos entre sí sea mayor sin detección de falsos positivos. Además, con el uso de un enfoque de agrupamiento

es posible detectar el mismo número o más de usuarios con una longitud menor de coeficientes/bits. Finalmente, de las 3 técnicas propuestas la técnica que utiliza la inserción de *frame* completo presenta una mayor robustez a todos los tipos de colusión.

## 6.2 Principales contribuciones

La principal contribución en este trabajo de tesis es un método de rastreo de copias de video el cual es robusto tanto a la pérdida de información por el proceso de compresión del estándar H.264/AVC como además de los diferentes ataques de colusión usados por los usuarios mal intencionados.

Otra contribución es la implementación de un sistema de conteo de votos a partir de una primera detección de identificador aplicado a videos.

Se realizó un análisis de la técnica *fingerprinting* propuesta mediante la inserción del identificador haciendo uso de bloques de tamaño $4 \times 4$, $8 \times 8$ y utilizando inserción de *frame* completo para los perfiles *Baseline* y *High* contemplando los tres tipos de *frame* existentes en el estándar H.264/AVC.

## 6.3 Limitaciones del enfoque propuesto

Debido a que la inserción se realiza *frame* por *frame* y un video de una película contiene un gran número de *frames*, el cual se define con el *framerate* y la duración de la película, el tipo de inserción sería muy grande para una aplicación en tiempo real como el video *streaming*.

## 6.4 Trabajo a futuro

Dado que la técnica propuesta con el uso de inserción mediante *frame* completo es independiente al proceso de compresión del estándar H.264/AVC, se puede validar la propuesta para otros esquemas de compresión con pérdida como lo es H.265 que si bien no tiene ahora un gran auge se perfila a ser ampliamente utilizado para los videos de alta resolución [16].

Dado que la inserción del identificador es *frame* por *frame* y que tanto la DCT como la IDCT son los procesos más costoso computacionalmente, es atractivo explorar la paralelización del método de inserción y detección del identificador del esquema propuesto.

Para una mayor detección de usuarios se puede estudiar la viabilidad de sintonizar los valores tanto $\beta_b$ y $\beta_u$, como $p_e$ y $p_{ec}$ para cada video y aún más específico para cada escena de video. Debido a que la naturaleza de cada video es diferente y ésta se puede considerar igual en cada escena.

# Bibliografía


[1] (1990). IEEE Standard Specifications for the Implementations of 8X8 Inverse Discrete Cosine Transform. *IEEE Std 1180-1990*.

[2] (2000). Information technology – Generic coding of moving pictures and associated audio information: Video. *ISO/IEC 13818-2:2000*.

[3] (2004). Information technology – Coding of audio-visual objects – part 2: Visual. *ISO/IEC 14496-2:2004*.

[4] Barg, A., Blakley, G. R., and Kabatiansky, G. A. (2003). Digital fingerprinting codes: problem statements, constructions, identification of traitors. *IEEE Transactions on Information Theory*, 49(4):852–865.

[5] Bhattacharya, S., Chattopadhyay, T., and Pal, A. (2006). A survey on different video watermarking techniques and comparative analysis with reference to H.264/AVC. In *2006 IEEE International Symposium on Consumer Electronics*, pages 1–6.

[6] Boneh, D. and Shaw, J. (1995). Collusion-secure fingerprinting for digital data. *Advances in Cryptology — CRYPT0' 95: 15th Annual International Cryptology Conference Santa Barbara, California, USA, August 27–31, 1995 Proceedings*, pages 452–465.

[7] Boneh, D. and Shaw, J. (1998). Collusion-secure fingerprinting for digital data. *IEEE Transactions on Information Theory*, 44(5):1897–1905.

[8] Chan, P. W., Lyu, M. R., and Chin, R. T. (2005). A novel scheme for hybrid digital video watermarking: approach, evaluation and experimentation. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(12):1638–1649.

[9] Cox, I., Miller, M., Bloom, J., Fridrich, J., and Kalker, T. (2008). *Digital Watermarking and Steganography*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2 edition.

[10] Cox, I. J., Kilian, J., Leighton, F. T., and Shamoon, T. (1997). Secure spread spectrum watermarking for multimedia. *IEEE Transactions on Image Processing*, 6(12):1673–1687.

[11] Danaher, B. and Waldfogel, J. (2012). Reel piracy: The effect of online film piracy on international box office sales. *Available at SSRN 1986299*, pages 1–28.

[12] Dominguez, H. D. J. O., Villegas, O. O. V., Sanchez, V. G. C., Casas, E. D. G., and Rao, K. (2014). The H.264 Video Coding Standard. *IEEE Potentials*, 33(2):32–38.

[13] Draft, I. (2003). Recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264| ISO/IEC 14496-10 AVC). *Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVTG050*, 33.

[14] encoding.com (2016). Global Media Format Report 2016. techreport, encoding.com. http://www.encoding.com/wp-content/uploads/FormatReport-2016-2.pdf, última vez visto 17-may-2017.

[15] Garcia-Hernandez, J. J., Feregrino-Uribe, C., and Cumplido, R. (2013). Collusion-resistant audio fingerprinting system in the modulated complex lapped transform domain. *PloS ONE*, 8(6):e65985.

[16] Grois, D., Marpe, D., Mulayoff, A., Itzhaky, B., and Hadar, O. (2013). Performance comparison of H.265/MPEG-HEVC, VP9, and H.264/MPEG-AVC encoders. In *2013 Picture Coding Symposium (PCS)*, pages 394–397.

[17] Hartung, F., Su, J. K., and Girod, B. (1999). Spread spectrum watermarking: malicious attacks and counterattacks. *Security and Watermarking of Multimedia Contents*, pages 147–158.

[18] He, S. and Wu, M. (2006). Joint coding and embedding techniques for multimediafingerprinting. *IEEE Transactions on Information Forensics and Security*, 1(2):231–247.

[19] He, S. and Wu, M. (2007). Collusion-resistant video fingerprinting for large user group. *IEEE Transactions on Information Forensics and Security*, 2(4):697–709.

[20] Hore, A. and Ziou, D. (2010). Image Quality Metrics: PSNR vs. SSIM. *20th International Conference on Pattern Recognition*, pages 2366–2369.

[21] Huynh-Thu, Q. and Ghanbari, M. (2008). Scope of validity of PSNR in image/video quality assessment. *Electronics Letters*, 44(13):800–801.

[22] Jain, R., Balooni, A., and Jain, H. (2015). Digital watermarking. *International Journal of Innovations and Advancement in Computer Science*, pages 184–191.

[23] Juurlink, B., Alvarez-Mesa, M., Chi, C. C., Azevedo, A., Meenderinck, C., and Ramirez, A. (2012). *Understanding the Application: An Overview of the H.264 Standard*, pages 5–15. Springer New York.

[24] Kalva, H. (2006). The h.264 video coding standard. *IEEE MultiMedia*, 13(4):86–90.

[25] Karmakar, A., Phadikar, A., Phadikar, B. S., and Maity, G. K. (2016). A blind video watermarking scheme resistant to rotation and collusion attacks. *Journal of King Saud University-Computer and Information Sciences*, 28(2):199–210.

[26] Kaur, R. (2016). A survey comparative analysis on video watermarking. *International Journal of Scientific Research in Science, Engineering and Technology IJSRSET*, 2(3):261–267.

[27] Kirovski, D. and Malvar, H. S. (2003). Spread-spectrum watermarking of audio signals. *IEEE Transactions on Signal Processing*, 51(4):1020–1033.

[28] Kuribayashi, M. (2011). Hierarchical Spread Spectrum Fingerprinting Scheme Based on the CDMA Technique. *EURASIP Journal on Information Security*, 2011(1):1–16.

[29] Kuribayashi, M. (2014). Countermeasure to non-linear collusion attacks on spread spectrum fingerprinting. In *2014 International Symposium on Information Theory and its Applications*, pages 50–54.

[30] Kush Amerasinghe (2009). H.264 for the rest of us. http://www.adobe.com/devnet/adobe-media-server/articles/h264_primer.html, última vez visto 20-abr-2017.

[31] Kutter, M. and Winkler, S. (2002). A vision-based masking model for spread-spectrum image watermarking. *IEEE Transactions on Image Processing*, 11(1):16–25.

[32] List, P., Joch, A., Lainema, J., Bjontegaard, G., and Karczewicz, M. (2003). Adaptive deblocking filter. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):614–619.

[33] Malvar, H. S. and Florêncio, D. A. (2003). Improved spread spectrum: A new modulation technique for robust watermarking. *IEEE Transactions on Signal Processing*, 51(4):898–905.

[34] Manaf, A. A., Boroujerdizade, A., and Mousavi, S. M. (2016). Collusion-Resistant Digital Video Watermarking For Copyright Protection Application. *International Journal of Applied Engineering Research*, 11(5):3484–3495.

[35] Marpe, D., Wiegand, T., and Sullivan, G. J. (2006). The H.264/MPEG4 Advanced Video Coding Standard and its Applications. *IEEE Communications Magazine*, 44(8):134–143.

[36] Mehta, S., Nallusamy, R., and Prabhakaran, B. (2016). Scene-based fingerprinting method for traitor tracing. *Multimedia Systems*, 22(2):197–211.

[37] Mirza, G. R. M. J. (2006). In-loop deblocking filter for H.264/AVC video. *Proceedings of the 5th WSEAS International Conference on Signal Processing, Madrid, Spain*, pages 235–240.

[38] Munoz-Hernandez, M. D., Garcia-Hernandez, J. J., and Morales-Sandoval, M. (2013). A collusion-resistant fingerprinting system for restricted distribution of digital documents. *PloS ONE*, 8(12):e81976.

[39] Nuida, K., Hagiwara, M., Watanabe, H., and Imai, H. (2007). Optimization of tardos's fingerprinting codes in a viewpoint of memory amount. *Information Hiding: 9th International Workshop*, pages 279–293.

[40] Richardson, I. E. (2011). *The H. 264 advanced video compression standard*. John Wiley & Sons.

[41] Ritter, G. X. and Wilson, J. N. (2000). *Handbook of Computer Vision Algorithms in Image Algebra*. CRC Press, Inc., Boca Raton, FL, USA, 2nd edition.

[42] Saadi, K. A., Bouridane, A., and Guessoum, A. (2014). H. 264/AVC digital fingerprinting based on spatio-temporal just noticeable distortion. *Fifth International Conference on Graphic and Image Processing. International Society for Optics and Photonics.*, 9069(2):1–6.

[43] Saxena, V. and Gupta, J. (2007). Collusion Attack Resistant Watermarking Scheme for Colored Images using DCT. *IAENG International Journal of Computer Science*, 34(2):1–6.

[44] Schaathun, H. G. (2014). Attacks on kuribayashi's fingerprinting scheme. *IEEE Transactions on Information Forensics and Security*, 9(4):607–609.

[45] Shahid, Z., Chaumont, M., and Puech, W. (2010). Spread spectrum-based watermarking for Tardos code-based fingerprinting for H.264/AVC video. In *2010 IEEE International Conference on Image Processing*, pages 2105–2108.

[46] Shahid, Z., Chaumont, M., and Puech, W. (2013). H. 264/AVC video watermarking for active fingerprinting based on Tardos code. *Signal, Image and Video Processing*, 7(4):679–694.

[47] Shaibu Akaeze, N. A. (2016). *Revenue Losses: Exploring Strategies Required by Managers to Inhibit Movie Piracy*. PhD thesis. Copyright - Database copyright ProQuest LLC; ProQuest does not claim copyright in the individual underlying works.

[48] Standard, J. (1992). Information technology-digital compression and coding of continuous-tone still images-requirements and guidelines. *International Telecommunication Union. CCITT recommendation*, 81:09.

[49] Su, K., Kundur, D., and Hatzinakos, D. (2002). A novel approach to collusion-resistant video watermarking. In *Proceedings of SPIE*, volume 4675, pages 491–502.

[50] Tardos, G. (2003). Optimal probabilistic fingerprint codes. In *Proceedings of the Thirty-fifth Annual ACM Symposium on Theory of Computing*, pages 116–125.

[51] Wagner, N. (1983). Fingerprinting. *IEEE Symposium on Security and Privacy*, pages 18–22.

[52] Wang, Z., Bovik, A. C., and Evan, B. L. (2000). Blind measurement of blocking artifacts in images. In *Proceedings 2000 International Conference on Image Processing*, volume 3, pages 981–984.

[53] Wang, Z., Sheikh, H. R., and Bovik, A. C. (2002). No-reference perceptual quality assessment of JPEG compressed images. In *Proceedings. International Conference on Image Processing*, volume 1, pages I–477–I–480.

[54] Wang, Z. J., Wu, M., Zhao, H. V., Trappe, W., and Liu, K. R. (2005). Anti-collusion forensics of multimedia fingerprinting using orthogonal modulation. *IEEE Transactions on Image Processing*, 14(6):804–821.

[55] Wiegand, T., Sullivan, G. J., Bjontegaard, G., and Luthra, A. (2003). Overview of the H.264/AVC video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):560–576.

[56] Wing, M. (2012). The digital copyright time bomb in the BRIC economies, some ideas from the UK for the Indian market. *International Journal of Law and Management*, 54(4):302–310.

[57] Xiph.org (2016). Xiph.org Video Test Media (derf's collection). *Xiph.org*. https://media.xiph.org/video/derf/, última vez visto 24-abr-2017.

[58] Zhang, X., Liu, Q., and Wang, H. (2012). Ontologies for intellectual property rights protection. *Expert Systems with Applications*, 39(1):1388–1400.

[59] Zhao, H. V., Wu, M., Wang, Z. J., and Liu, K. R. (2005). Forensic analysis of nonlinear collusion attacks for multimedia fingerprinting. *IEEE Transactions on Image Processing*, 14(5):646–661.