UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA
ELÉCTRICA – TELECOMUNICACIONES

SISTEMA DE IDENTIFICACIÓN DE ESCENAS EN JPEG2000 Y SUS APLICACIONES AL INDEXADO MPEG7

T E S I S

QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN INGENIERÍA

PRESENTA:
SENYAZEN REYES MEZA

TUTOR PRINCIPAL
DR. VÍCTOR GARCÍA GARDUÑO, FACULTAD DE INGENIERÍA

MÉXICO, D. F. AGOSTO DE 2013

JURADO ASIGNADO:

Presidente: Dr. Francisco García Ugalde

Secretario: Dr. José María Matías Maruri

Vocal: Dr. Víctor García Garduño

1er Suplente: Dr. Bohumil Psenicka

2do Suplente: Dr. Javier Gómez Castellanos

Lugares en donde se realizó la tesis:

Universidad Nacional Autónoma de México, México, D.F.
Laboratoire Bordelais de Recherche en Informatique, Francia, Bordeaux.

TUTOR DE TESIS

Dr. Víctor García Garduño

Agradecimientos

A mi familia, especialmente a mis padres y hermana Kaory, por todo el apoyo incondicional que me han brindado a lo largo de mi vida. A mi novio, casi esposo, Felipe, por siempre estar al pie del cañón y nunca dejarme sola en la embarcación.

A la UNAM, bondadosa universidad que me ha abierto innumerables puertas del conocimiento y oportunidades profesionales. Por solventar y fomentar mis estudios de Posgrado mediante una beca.

Al Dr. Víctor García Garduño, por ser admirable guía profesional en mi caminar. Por confiar en mí y haber impulsado la estancia en el extranjero. Por su tiempo, atención, paciencia y apoyo para conmigo.

A la Professeur Jenny Benois-Pineau por el apoyo solidario durante la estancia en Francia.

Sistema de identificación de escenas en JPEG2000 y sus aplicaciones al indexado MPEG7

Resumen

El objetivo de este trabajo de tesis es proponer un sistema de identificación de escenas, definiendo como escena aquellos fotogramas que compartan características visuales en un mismo contexto, utilizando una cámara capaz de capturar video. Una vez capturado el video, el sistema selecciona algunos frames, de los cuales se extraen descriptores basados en color, textura y forma, permitiendo efectuar una clasificación de las escenas más relevantes del video. Para ello se hace uso del estándar JPEG2000 consiguiendo de esta forma comprimir dichas imágenes y al mismo tiempo obteniendo información valiosa de las imágenes, es decir, características fundamentales que nos ayudan a describirlas. Dentro de éste contexto, el trabajo es basado en el dominio de la Transformada Discreta de Wavelet-9/7 de Daubechies, la cual proporciona información de contorno. Seguido de ello, se calcula el histograma de color (YC_rC_b) a cada imagen Wavelet, principal descriptor de nuestras imágenes. Contando ya con información descriptiva, se realiza un clustering de tipo jerárquico para obtener las escenas principales del video, en donde cada escena será representada por una imagen clave (center cluster), la cual representará al conjunto de imágenes características de dicha escena, utilizando una distancia del tipo Chi-square (kernel) para medir la similitud entre escenas, es decir, la comparación de histograma a histograma de cada una de las imágenes Wavelets. Contando con las imágenes que mejor representan cada una de las escenas del video. A partir de éste análisis, se proponen algunas aplicaciones al indexado MPEG7, tal como lo es la identificación de imágenes a través de dispositivos móviles, para la cual se piensa en una aplicación programada en un dispositivo celular que permita capturar un video de mediana calidad de objetos. Ésta información más los datos GPS (Global Positioning System) del usuario, son enviados a un servidor para fines de identificación, para después ofrecerle a dicho usuario una breve descripción de los principales escenarios que se encuentran en su entorno, esto último mediante el uso de SVM's (Support Vector Machines).

Palabras clave: JPEG2000, wavelet, histograma de color, clustering jerárquico, MPEG7, SVMs.

Índice

Introducción

El video digital hoy en día desempeña un mayor papel en nuestra vida cotidiana, tal como lo son aplicaciones que van desde noticias, entretenimiento, investigación científica, seguridad y vigilancia. Actualmente las cámaras y medios de almacenamiento son cada vez menos costosos, los teléfonos celulares poseen poderosas herramientas, los cuales se han convertido en una fuente importante en lo que a la búsqueda web se refiere. En la mayoría de las búsquedas web se trata de consultas de tipo texto, sin embargo a veces no siempre resulta adecuada la descripción de la búsqueda en tan sólo palabras, además de que la introducción del texto en dispositivos pequeños resulta ser un inconveniente. Contando con una imagen y datos GPS, una simple búsqueda puede potencializarse, pues está nos proporciona información más específica y certera, lo cual resulta de gran utilidad para conocer a mayor detalle el entorno en el que nos encontramos. Esto ha dado lugar a una gran producción de contenido de vídeo como antes nunca, lo que requiere el desarrollo de métodos de indexación y algoritmos de recuperación de datos de vídeo mayormente eficientes. Hoy por hoy existen un gran número de técnicas dentro del estado del arte que trabajan en la indexación de video de acuerdo con el contenido global de la escena, tal como el color, la textura, el brillo, etc..

La utilización de imágenes para identificación y clasificación de objetos en general está muy difundido existiendo dos estrategias principales, la primera basada en la extracción de características (forma, tamaño, etc.) y la segunda, más en boga en el último tiempo, busca el reconocimiento de patrones dentro de la imagen. Para este último paradigma, existen ejemplos que han demostrado buenos resultados, desde las redes neuronales, hasta las más novedosas máquinas de soporte vectorial (Support Vector Machines-SVMs).

El trabajo aquí presentado, plantea el reconocimiento de entornos a través de la indexación de vídeos, en donde la extracción automática de objetos de interés es una etapa esencial del análisis, la indexación y de igual forma de la compresión de video. En el caso del análisis y la indexación se describe en la organización automática de contenido [5]. Dentro del caso de la compresión, es definir las zonas a codificar con mayor precisión y conforme al estándar JPEG2000 [6] y en regiones de interés. La segmentación de color contenida en el dominio

Wavelet se centra esencialmente en la sub-banda LL [27,28]. Se propone usar solamente la sub-banda LL, con el fin de agrupar en una sola imagen (means cluster) aquellas imágenes que pertenezcan a la misma escena.

El trabajo se organiza de la siguiente manera:

Introducción

Este apartado presenta los fundamentos y motivaciones del proyecto de tesis, en el panorama, la importancia y la necesidad que tienen la utilización de imágenes para la clasificación de las mismas, reconocimiento de entornos e incluso, clasificación de objetos, para con nuestra vida diaria.

Capítulo 1- Estado del Arte

Este capítulo constituye la visión general de los estándares y los más recientes métodos utilizados en la descripción de contenidos de información visual, centrándose en los estándares MPEG7 y JPEG2000.

Capítulo 2- Aplicación del estándar JPEG2000-representación de los datos

En esta sección se describen los principales descriptores de los frames, la Transformada Discreta de Wavelet y el histograma de color en baja frecuencia, los cuales proporcionaran las características fundamentales y relevantes de las imágenes, elementos decisivos para la posterior comparación y clasificación de las mismas.

Capítulo 3- Clasificación de imágenes

Detalla los algoritmos empleados para el agrupamiento del total de los frames en únicamente k imágenes clave (Cluster Jerárquico).

Capítulo 4- Aplicaciones al indexado MPEG 7

Éste capítulo ejemplifica algunas aplicaciones existentes respecto al indexado MPEG 7, así como la descripción de una propuesta de aplicación móvil para la identificación de imágenes y datos GPS.

Capítulo 5- Evaluación y resultados del agrupamiento.

Muestra la explicación de los resultados obtenidos de acuerdo a las distintas pruebas y criterios realizados, así como una comparativa de los resultados alcanzados.

Capítulo 6- Conclusiones y perspectiva

Se presentan las conclusiones alcanzadas tras la realización del proyecto y en análisis a conciencia de los resultados obtenidos. Por otra parte se presentan las posibles líneas de investigación respecto al trabajo realizado.

Referencias

Capítulo 1

Estado del Arte

La indexación de información existe desde hace siglos. En un principio la indexación se realizaba manualmente, pero con la llegada de la digitalización y el aumento considerable de archivos multimedia se hace necesario la creación de estándares de compresión. Con dicho propósito surge MPEG, un grupo de trabajo de ISO/IEC que se encarga del desarrollo internacional de normas para el procesado, compresión, descompresión y codificación tanto de vídeo como de audio. El primer estándar que surgió fue MPEG-1, destinado a la compresión de video y audio, y de donde nacieron las normas dedicadas a los CD's de video y al formato MP3. Dos años después ya estaba en el mercado un conjunto de nuevas normas, MPEG-2, responsable de definir las pautas para el servicio de TV por satélite, por cable y para formato DVD. Posteriormente se presenta el tercer estándar denominado MPEG-4, el cual proporciona un cambio muy positivo en la línea de interpretación de vídeos de las normas anteriores. Hasta este momento todos los estándares expuestos compartían una similitud, representaban con bits el contenido del material audiovisual en sí mismo. Pero a partir de aquí esta situación avanza cuando surge uno de los estándares más innovadores de MPEG. Con MPEG-7 la filosofía es distinta. El objetivo ahora es codificar los datos que describen la información. Introduciendo de este modo el concepto de metadato. Los contenidos pueden ser descritos de distintas formas dependiendo de la necesidad, ya que las características descriptivas deben tener un significado en el contexto de la aplicación. Estas descripciones deberán ser distintas para distintos dominios de usuarios y sistemas. Esto significa que no se puede generar un sistema único para la descripción de contenidos, sino que se tendrán que proveer un conjunto de métodos y herramientas para satisfacer los distintos puntos de vista que distintos usuarios pueden tener. El material multimedia, pues, puede ser descrito usando distintos niveles de

abstracción. Cuanto mayor sea dicho nivel de abstracción, más difícil es efectuar un proceso automático.

1.1 El estándar MPEG-7

El estándar MPEG-7, conocido como Interfaz de Descripción de Contenido Multimedia, proporciona un amplio conjunto de herramientas estándar para describir contenido multimedia, tanto para usuarios humanos como para sistemas automáticos que procesen información audiovisual. Estas herramientas de descripción, conocidas como Description Tools, que son los elementos de metadatos y su estructura y relaciones, definidas en el estándar como Descriptores (D) y Esquemas de Descripción (DS), sirven para crear descripciones que son la base para aplicaciones que permitan éste necesario acceso eficiente a contenido multimedia. Es una tarea complicada dado el amplio espectro de requisitos y aplicaciones multimedia que pretende abarcar, así como el gran número de características audiovisuales de importancia en este contexto.

Las herramientas de descripción que ofrece MPEG-7 son, por tanto, independientes de cómo está almacenado o codificado el contenido (se puede trabajar con señales analógicas o digitales). MPEG-7 parte de la base de que existe una representación (estandarizada o no) del contenido y se encarga de describir todo lo relacionado con él.

1.1.1 Características de MPEG-7

Las características principales que describen MPEG-7 son las siguientes [11]:

- El tipo de dato audiovisual considerado va desde el audio, la voz, imágenes fijas, video, mallas en 3D y gráficos hasta la información de cómo los objetos están distribuidos en las escenas.
- Las descripciones del material multimedia pueden ser de dos tipos:
- Información sobre el contenido, como puede ser el autor, el género, el título o el formato.
- Información existente en el contenido, la cual permite describir el elemento a través de significado semántico (descripción de alto nivel), relacionado con la interpretación

del contenido o significado estructural (descripción de bajo nivel) el cual permite la extracción automática de color, forma texturas, sonidos, etc.

- Los descriptores de alto nivel se caracterizan por ser eficientes y directos pero poco flexibles mientras que los de bajo nivel son genéricos, flexibles y permiten búsquedas inteligentes [12].
- La descripción es independiente del formato que tengan los datos audiovisuales. Las normas definidas no dependen de cómo esta codificada o almacenada la información. MPEG-7 no solamente describe contenidos codificados en MPEG sino que es capaz de describir desde un video en formato VHS hasta una imagen impresa en un papel.
- Aprovechando las ventajas de MPEG-4, este nuevo estándar ofrece la posibilidad de hacer descripciones referentes a partes más específicas del material. Pueden ser referidas a objetos o las relaciones temporales y espaciales dentro de la escena.
- Uno de los puntos fuertes del estándar se centra en las posibilidades que ofrece en el momento de transmitir las descripciones. MPEG-7 permite multiplexarlas con el contenido, con la ventaja de tenerlas físicamente asociadas al material, o transmitirlas por separado junto con un puntero el cual señala el objeto fuente [12].
- Tanto la generación de descripciones, la cual engloba la extracción de características, indexación y segmentación, como la consumición de dichas descripciones gracias a máquinas de búsqueda son aspectos no considerados por las normas MPEG-7.

En el esquema de la parte inferior se puede observar una aplicación real de MPEG-7 donde los recuadros de color café corresponden a los procesos que se pueden realizar con libertad, ya que no están definidos por el estándar, mientras que el recuadro central de color azul es donde se aplican las normas estandarizadas [12].

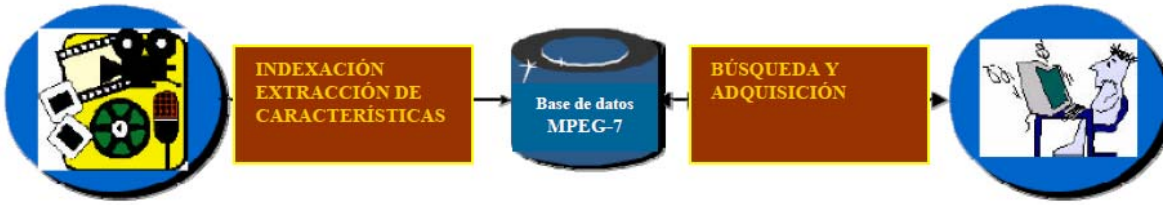


Figura 1.1 Aplicación real de MPEG-7[1]

[1] Imagen tomada de las notas de clase "Introducción al estándar MPEG-7, Universidad Politécnica de Cataluña, 2006.

MPEG-7 utiliza XML *Schema* (XML, *Extensible Markup Language*) como lenguaje para la representación textual del contenido, lo cual le permite ser flexible y aumentar las herramientas de descripción existentes.

El alcance de MPEG-7 se limita al formato de las descripciones, y no estandariza ni la producción de esas descripciones ni el uso que se va a dar.

1.1.2 Constitución de MPEG-7

De entre las partes de que consta MPEG-7 las más importantes son [11]:

- *MPEG-7 Systems*: arquitectura del estándar y herramientas necesarias para preparar las descripciones de MPEG-7 para su transporte y almacenamiento eficiente.
- *MPEG-7 Description Definition Language*: lenguaje para definir nuevos DSs y, quizás eventualmente, también para nuevos Ds.
- *MPEG-7 Audio*: proporciona estructuras para describir material sonoro. En ellas se tiene un conjunto para de descriptores de bajo nivel (para características espectrales, temporales, etc.), y otro de herramientas de alto nivel.
- *MPEG-7 Visual*: estructuras básicas y descriptores que cubren las siguientes características visuales básicas: color, textura, forma, movimiento, localización y reconocimiento facial.
- *MPEG-7 Multimedia Description Schemes*: elementos (DSs y Ds) que describen información genérica, es decir, que no es ni puramente visual ni de audio.

1.1.3 Esquemas de descripción multimedia

Los Esquemas de Descripción Multimedia (Multimedia Description Schemes) son estructuras de metadatos para describir y anotar contenido audiovisual [11]. Estos DSs proporcionan una modelo estándar para describir en XML aspectos importantes relacionados con la descripción de contenido audiovisual y gestión de contenido, para facilitar las tareas de búsqueda, indexado, filtrado y acceso a dicho contenido. Estos esquemas de descripción se definen utilizando el lenguaje descrito en MPEG-7 DDL. Las descripciones MPEG-7 resultantes pueden ser expresadas bien en formato textual, legible por personas, y útil para búsqueda, edición y

filtrado, o bien comprimidas en formato binario, más adecuado para almacenamiento y transmisión.

1.2 Descriptores visuales de MPEG-7

Los descriptores visuales describen las características visuales de los contenidos dispuestos en imágenes o en vídeos. Describen características elementales tales como la forma, el color, la textura o el movimiento, entre otros.

1.2.1 Descriptores de color

1.2.1.1 Color Space

Consiste en un tipo de datos que especifica el espacio de color en el cual se expresan o trabajan los otros descriptores de color. Los espacios de color que contempla son los siguientes:

- RGB (Red, Green, Blue)
- YCrCb (Luminancia, Crominancia)
- HSV (Hue, Saturation, Value)
- HMMD (Hue, maximum, minimum, difference)
- Matriz de transformación lineal con referencia a RGB
- Monocromo

1.2.1.2 Color Quantization

Este descriptor define una cuantificación uniforme de un espacio de color determinado. El número de valores que el cuantificador produce es configurable de manera que posee una flexibilidad elevada que le da una amplia gama de usos. Dentro de MPEG-7 este descriptor se

combina con descriptores del color dominante, para hacer comparables por ejemplo dos resultados de un determinado descriptor.

1.2.1.3 Dominant Color(s)

Este descriptor es el más conveniente para ser utilizado en imágenes o zonas de ellas, en las cuales un pequeño número de colores es suficiente para caracterizar la información cromática de la región determinada. Sería aplicable por ejemplo en imágenes de banderas o marcas determinadas. En este caso la cuantificación se usa para extraer un reducido número de colores que sean suficientes como para caracterizar la imagen o región. También se calcula una coherencia espacial entre estos colores y dónde están situados la cual cosa se utilizará en algoritmos de similitud.

1.2.1.4 Scalable Color

Este descriptor consiste en un histograma de color en el espacio HSV, codificado con una medida de Haar. Su representación se puede escalar de manera que se adecue lo máximo al tamaño de datos con el que se quiere trabajar. Este descriptor es útil en comparaciones imagen-imagen o en búsquedas basadas en características de color. La fiabilidad de la búsqueda aumenta proporcionalmente al número de colores distintos que se tengan en cuenta en el histograma.

1.2.1.5 Color Layout

Este descriptor permite representar la distribución espacial del color dentro de las imágenes de una manera muy compacta, con lo cual representa una herramienta de gran utilidad a la hora de buscar imágenes a partir de modelos determinados, y lo hace con gran eficiencia y velocidad. Su fácil cálculo permite también usarlo en la comparación de secuencias de imágenes, en las cuales se precisa un análisis de similitud entre cada una de sus componentes.
Las grandes ventajas de este descriptor son:

- No depende ni del formato, ni de la resolución, ni del margen dinámico de las imágenes o videos en que se use. Por este motivo, puede usarse para comparar imágenes o videos con distintas resoluciones o para comparar imágenes enteras con partes de imágenes.
- El software/hardware que requiere es relativamente mínimo (usa solamente 8 bytes por imagen cuando trabaja por defecto). Esto lo convierte en un descriptor adecuado para ser utilizado en dispositivos móviles en los que los posibles recursos se ven limitados por la capacidad del hardware.
- Las características que analiza de una imagen las representa en dominio frecuencial de manera que los usuarios pueden manipular con facilidad su contenido ciñéndose a la sensibilidad del sistema visual humano.
- Permite trabajar con distintas precisiones de descripción de manera que se agudizan las comparaciones cuando sea necesario.

1.2.1.6 Color Structure Descriptor

Este descriptor caracteriza la distribución de los colores en una imagen. Construye una especie de histograma de color, en el cual tendrán mayor importancia a los colores que más se reparten por la imagen. El descriptor divide la imagen en bloques de 8x8 píxeles y analiza dentro de estos bloques los distintos colores que aparecen, incrementándolos así en el histograma. A diferencia de un histograma de color, permite distinguir entre dos imágenes que tengan la misma cantidad de píxeles de un color pero con distinta distribución de estos píxeles. Este descriptor es útil para comparaciones imagen-imagen y añade funcionalidades distintas a las del histograma de color que permiten mejorar la búsqueda de similitud en determinados tipos de imágenes, como por ejemplo las imágenes de naturaleza.

1.2.1.7 GoF/GoP Color (Group of Frames/Group of Pictures)

Este descriptor es una extensión del Scalable Color, que a diferencia de éste, que está definido para imágenes inmóviles, se aplica a secuencias de video o secuencias de imágenes fijas. Este descriptor da la posibilidad de calcular de tres formas distintas el histograma de color:

- Histograma promedio: toma de cada imagen de la secuencia el promedio de los valores del histograma.
- Histograma de mediana: toma de cada imagen de la secuencia el valor central del conjunto de valores del histograma. Es más fiable ante errores o picos de intensidad de la imagen.
- Histograma de intersección: toma de cada imagen de la secuencia el mínimo del conjunto de valores del histograma, para así ver cuál es el color "menos común" en el conjunto de imágenes.

1.2.2 Descriptores de textura

1.2.2.1 Homogeneous Texture

Este descriptor emergió como una importante herramienta a la hora de buscar y escoger dentro de grandes colecciones de imágenes de gran similitud visual. Este descriptor utiliza un banco de 30 filtros que permite obtener una afinada descripción de las distintas texturas de la imagen para poder comparar de esta manera con las de otras. Es una herramienta muy útil por ejemplo para distinguir determinadas zonas en imágenes aéreas, por ejemplo, cultivos.

1.2.2.2 Texture Browsing

Este descriptor especifica la caracterización perceptiva de una textura, la cual es similar a la caracterización de ella que hace un ojo humano, en cuanto a términos de regularidad, tosquedad y direccionalidad. Es útil para búsquedas y clasificaciones a groso modo de texturas. Su implementación es parecida a la del anterior.

1.2.2.3 Edge Histogram

El Edge Histogram es un descriptor que nos facilita información sobre el tipo de contornos o bordes que aparecen en la imagen. Trabaja dividiendo la imagen en 16 sub-imágenes y es

capaz de analizar en ellas el tipo de bordes existentes con el uso de distintos filtros que le permiten diferenciar si son bordes horizontales, verticales, oblicuos o aleatorios. Su utilización principal es la comparación imagen-imagen, especialmente en imágenes de naturaleza con una gran no uniformidad de contornos. Su uso es muy útil también en combinación con el de otros descriptores como por ejemplo el histograma de color.

1.2.3 Descriptores de forma

1.2.3.1 Region Shape

La forma de un objeto en una imagen, puede consistir en una sola región o un conjunto de regiones como vemos en las siguientes imágenes:

Éste descriptor permite clasificarlas según esta característica, de manera que podemos comparar formas de distintas imágenes y ver por ejemplo si se trata del mismo objeto u objetos similares.

Figura 1.2 Ejemplos de imágenes con las que trabaja Region Shape

Las grandes ventajas de este descriptor son su reducido tamaño y su velocidad, hay que tener en cuenta que el tamaño de los datos necesarios para su representación está fijado en 17.5 bytes.

1.2.3.2 Contour Shape

A diferencia del anterior, este descriptor en lugar de analizar el conjunto de regiones que dan lugar a una forma, relaciona ésta última con su contorno. Se caracteriza por representar muy bien las características de contorno con lo que facilita posteriores búsquedas y

recuperaciones, es robusto ante movimientos, ante oclusiones en las formas y ante distintas perspectivas, y es sumamente compacto.

1.3 El estándar JPEG2000

En 1996 el comité JPEG (Joint Photographic Experts Group) comenzó a investigar un nuevo estándar de codificación de imágenes con los últimos avances. A este proyecto se le llamó JPEG2000, el cual fue dividido en seis partes, posteriormente ampliado a once.

La primera constituye el bloque principal del estándar. El resto lo forman extensiones, aplicación al movimiento, testeo, implementación de software y formación del formato de archivo, seguridad, etc.. Las fechas previstas para las seis primeras partes estaban comprendidas entre el 2000 y 2002, de ahí el nombre adoptado. Esta parte del proyecto se basa en dicha primera parte para la implementación de una aplicación software. Para ello, antes conviene conocer cuáles son las características del estándar JPEG2000 en una sencilla pero completa descripción.

El comité JPEG se formó en 1986 bajo los regímenes de la ISO y la ITU-T. El primer estándar que publicó fue el ya clásico JPEG que tuvo una gran aceptación por su sencillez, rapidez y eficacia, lo cual contribuyó a su amplia difusión, aunque probablemente Internet y su libre distribución hayan tenido también mucho que ver. Sus principales características eran: (i) El uso de la Transformada Discreta del Coseno (DCT), basada en codificación con pérdidas y códigos Huffman con la restricción de 8 bpp (imagen de color) de entrada; (ii) Una extensión para controlar diferentes realces de la imagen; y (iii) Un modo sin pérdidas con codificación predictiva y códigos Huffman o aritméticos.

Pero pronto las cualidades del estándar JPEG serían insuficientes en cuanto a calidad de compresión, como los típicos artefactos de la DCT, como la flexibilidad, en cuanto a formación de la trama final. Con estas premisas no queda más remedio que abandonar la DCT por mejores transformaciones que permitan aplicar mejor los métodos perceptuales conocidos, como el enmascaramiento de coeficientes, la sensibilidad al contraste (CSF) o la selección de zonas de mejor calidad; y transformadas que permitan mayor flexibilidad en la organización de la trama para poder elegir mejor la tasa final de compresión, introducir códigos de corrección de errores o simplemente por elegir en qué orden se codificará la imagen. Esto

será fundamental para abarcar un gran número de aplicaciones. Aplicaciones como compresión de imágenes médicas, librerías digitales, multimedia, internet o teléfonos móviles. Los objetivos técnicos eran por tanto mejorar la calidad de las imágenes tratadas y aumentar las posibles aplicaciones. Así, las características se pueden resumir en:

- Eficiencia de compresión.
- Compresión con y sin pérdidas.
- Representación con resolución múltiple.
- Decodificación SNR progresiva y en resolución.
- Fragmentación de la imagen.
- Regiones de Interés.
- Protección frente a Errores.
- Acceso y procesamiento aleatorio de la trama.
- Tratamiento post-compresión.
- Posibilidad de interacciones en la codificación.
- Un formato más flexible.

Detrás de todo esto está, como con el anterior JPEG, el crear un estándar libre de distribución que no suponga un coto privado como ocurre con otros algoritmos, aunque evidentemente las empresas colaboradoras siempre tendrían beneficios directos y/o indirectos derivados de esta actividad.
En 1997 se propusieron 20 algoritmos y fue la descomposición Wavelet la que se adoptó como base del nuevo estándar. Actualmente la transformada Wavelet está extendida a innumerables e incluso insospechados campos. El éxito de esta derivó entonces del estado del arte que se tenía por entonces y de las posibilidades que permitía.

1.3.1 Características principales

El diagrama de bloques típico del JPEG2000 está constituido por una etapa de pretratamiento de la imagen, una transformada discreta wavelet, una etapa de cuantización, una codificación aritmética (también llamada tier-1) y después la organización de trama (tier-2). Posteriormente se irá desgranando con más detalle cada una de las partes. Ver figura 1.3

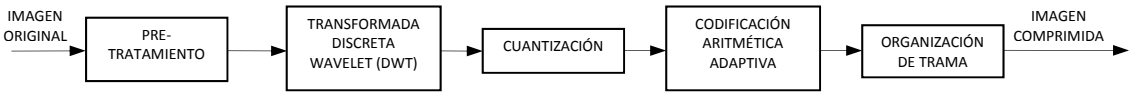


Figura 1.3 Diagrama de bloques JPEG2000

1.3.1.1 Etapa de pre-tratamiento

Esta etapa constituye la primera toma de contacto entre la imagen y el algoritmo. Este último examina las características de aquella, adaptándose y moldeándola. Las características serán el valor de offset, el rango de la imagen y el número de componentes.
El valor de offset es un escalado de la forma:

$$I_e = I_O - O_{DC} \tag{1.1}$$

Donde I_e e I_o son la imagen escalada y original respectivamente
O_{DC} está representado por:

$$O_{DC} = 2^{B_I - 1} \tag{1.2}$$

Donde B_I es el número de bits de la imagen original.
Para hacer la transformación inversa basta con sumar este valor de offset. El rango dinámico quedará de la forma $[-2^{B_I-1}, 2^{B_I-1} - 1]$.

Finalmente si la imagen tiene varias componentes, de color se entiende, se pueden utilizar dos tipos de transformaciones a partir de las componentes originales Rojo-Verde- Azul RGB: Una transformación con pérdidas llamada Transformación Irreversible de Color (ITC), que como su propio nombre indica, no se puede realizar la transformación inversa sin asumir unas pérdidas. La transformación se realiza mediante la multiplicación matricial:

$$\begin{pmatrix} Y \\ C_r \\ C_b \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.16875 & -0.33126 & 0.5 \\ 0.5 & -0.41869 & -0.08131 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{1.3}$$

La transformación ITC inversa es:

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1.0 & 0 & 1.402 \\ 1.0 & -0.34413 & 0.71414 \\ 1.0 & 1.772 & 0 \end{pmatrix} \begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix} \tag{1.4}$$

La otra transformación es la llamada *Transformación Reversible de Color (RTC)*, que puede ser usada para casos con o sin pérdidas. Esta se define como:

$$Y = \left\lfloor \frac{R+2G+B}{4} \right\rfloor, U = R - G \,, V = B - G \qquad (1.5)$$

Donde $\lfloor w \rfloor$ es el entero más alto que es igual o más pequeño que w. La transformación RTC inversa es:

$$G = Y - \left\lfloor \frac{U+V}{4} \right\rfloor, R = U + G \,, B = V + G \qquad (1.6)$$

La base de las dos transformaciones es obtener un canal de luminancia y dos de diferencia de color, diferencia de rojo y diferencia de azul, con lo que se trata la imagen con tres canales por separado con diferentes características de codificación. Evidentemente tanto el valor de offset como el tipo de transformación elegida deben ser conocidos por el decodificador. Otra operación en la codificación de color es una reindexación de la paleta de colores donde los saltos dentro de la paleta son relativos y los colores más frecuentes se ponen en zonas más cercanas. A partir de aquí la descripción se centrará en la componente de luminancia ya que el tratamiento para las otras dos es similar.

1.3.2. Transformada de Wavelet

En este punto haremos una pequeña descripción de la Transformada Wavelet Discreta, pero más adelante dará una descripción más detallada. La Transformada Wavelet constituye una de las características claves de innumerables algoritmos de compresión y en concreto del estándar JPEG2000. La Transformación Wavelet (WT) supone un nuevo enfoque en el tratamiento de la señal, y en concreto del tratamiento de imágenes [11]. El objetivo de toda transformación, en cuanto a compresión se refiere, es representar la información en otro espacio de una manera más eficiente. La base de dicho espacio será la clave de su eficacia.

1.3.2.1 Transformada de Wavelet Discreta

La DWT (*Discrete Wavelet Transform*) es la transformada que ha sustituido a la DCT (*Discrete Coseno Transform*) en el nuevo estándar JPEG. Las características que han incluido en esta decisión, como se comenta en los anteriores párrafos, son la representación multirresolución, la localización y la capacidad de realizar una codificación sin pérdidas con filtros enteros. Por

simplicidad a continuación se tratará la DWT de una dimensión, ya que la extensión a las dos dimensiones es inmediata.

Efectuar el análisis de ondas en función de una imagen conlleva un análisis multiresolución. Mallat [14] propone dicho cálculo haciendo uso de bancos de filtros. Los bancos de filtros utilizados aquí son asociados dos filtros por una dimensión: un filtro paso bajas (La) y un filtro paso altas (Ha). Recordando que filtrar un vector unidimensional (en este caso cada fila y columna de la imagen) resulta de la convolución de ese vector con la respuesta impulso discreta y filtros finitos. Así pues la convolución $\hat{I}$, de una línea I_l con el filtro paso bajas H de longitud T, está dada por:

$$\hat{I}_l = I_l * H \quad ; \quad \hat{I}_l(t) = \sum_{-T/2}^{T/2} I_l(t-\tau)H(\tau) \tag{1.7}$$

En primer lugar las líneas de la imagen son procesadas por el filtro paso-bajas y el filtro paso-altas. Para obtener los coeficientes de transformación de ondas, estos filtros son aplicados separadamente a las filas y a las columnas de la imagen. Esto da como resultado un número de coeficientes 2 veces más que en la imagen original, para lo que la imagen de cada filtro es submuestreada(↓) por un factor de 2. Las columnas son filtradas (paso bajo- paso alto) y submuestreadas a continuación. Esta combinación de filtrado da como resultado 4 imágenes o sub bandas denotadas como LL1, LH1, HL1 y HH1 que forman el primer nivel de descomposición Wavelet.

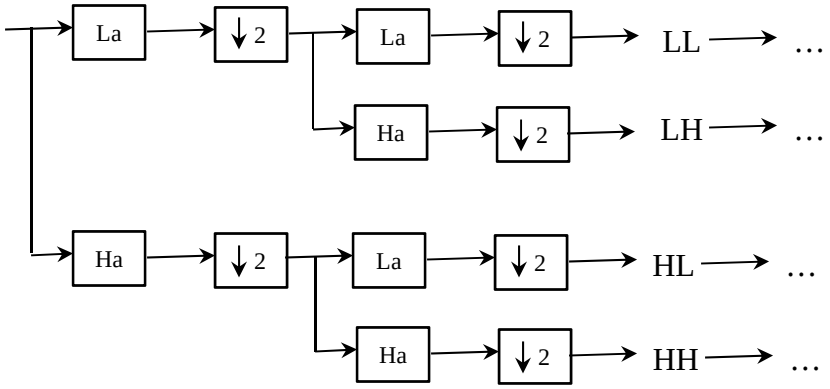


Figura 1.4 Análisis de la Transformada Discreta de Wavelet

Típicamente éste proceso es repetido K veces, resultando así $W_t = \{LL^0, LL^k, LH^k, HL^k, HH^k, k = 1 \dots K\}$, dónde k es el nivel de descomposición. W_t es la trama de onda en el instante t correspondiente a la imagen I_t en un instante t del video. Cuando k=0 la imagen es la resolución completa, entendiendo como la subbanda LL^0, y cuando k=K se tiene la resolución mínima de la imagen.

Los filtros que se utilizan, elegidos básicamente por sus buenas propiedades, son el filtro de coeficientes enteros (5,3) para las transformadas reversibles, codificación sin pérdidas, y el filtro de Daubechies (9,7) para las transformadas irreversibles, codificación con pérdidas.

Filtro de Análisis		
i	Filtro paso-bajas L_a	Filtro paso-altas H_a
0	0.6029490182363579	1.115087052456994
$\pm$1	0.2668641184428723	-0.591271763114247
$\pm$2	-0.07822326652898785	-0.05754352622849957
$\pm$3	-0.01686411844287495	0.09127176311424948
$\pm$4	0.02674875741080976	
Filtro de Síntesis		
i	Filtro paso-bajas L_a	Filtro paso-altas H_a
0	1.115087052456994	0.6029490182363579
$\pm$1	-0.591271763114247	0.2668641184428723
$\pm$2	-0.05754352622849957	-0.07822326652898785
$\pm$3	0.09127176311424948	-0.01686411844287495
$\pm$4		0.02674875741080976

TAB. 1 Coeficientes del filtro de ondas 9/7 de Daubechies

La sub-banda LL:

- Contiene una imagen reducida respecto a la imagen normal

La sub-banda HF

- Contiene la información de contorno
- LH: contornos horizontales
- HL: contornos verticales
- HH: contornos diagonales

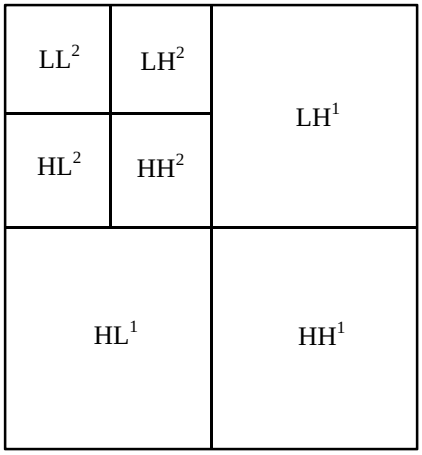


Figura 1.5 Notación de la descomposición mediante la Transformada Discreta de Wavelet

1.3.3 Cuantización

El antiguo estándar JPEG utilizaba un cuantificador uniforme con una reconstrucción en la mitad del intervalo. El ancho de cuantificación era variable en función de la sensibilidad para adaptar la percepción al SVH y después se aplica un mapa de dimensiones 8x8 y 1 byte por cada uno.

Una importante diferencia es la presencia de una *zona muerta* (elimina ruido y aumenta la frecuencia de ceros, lo que facilita la codificación posterior) en el cuantificador. La razón de incluir esta zona se debe a que el cuantificador óptimo para una señal continua con distribución Laplaciana, como es el caso de la DCT y DWT, es precisamente el que incluye dicha zona muerta [7]. El tamaño de la zona disminuye a medida que la varianza de la distribución Laplaciana aumenta. Este tamaño suele ser menos que 2 veces el ancho de la cuantificación, Δ_b, y típicamente alrededor de 1.

La ventaja de un cuantificador con zona muerta del doble de Δ_b es que el resultado de la codificación es el mismo tanto si se trunca el valor de un cuantificador de M_b bits en el bit N_b como si se cuantifica el mismo valor con un cuantificador con un ancho $\Delta_b = 2^{M_b - N_b}$ donde M_b es la magnitud.

Esto implica que la reconstrucción va a ser correcta si se truncan los valores obtenidos, lo que permite alcanzar las tasas deseadas sin necesidad de hacer varias iteraciones como era el caso de JPEG.

Los parámetros a elegir en el cuantificador serán el ancho de cuantificación y la zona muerta en relación con el anterior donde la elección puede ser en base a consideraciones perceptuales o simplemente en base a la tasa de compresión final.

La cuantificación viene dada por:

$$q_b(u,v) = sign\big(I_b(u,v)\big)\left[\frac{|I_b(u,v)|+nz*\Delta_b}{\Delta_b}\right], nz \in (-1,1) \qquad (1.8)$$

Siendo I_b la banda correspondiente de la imagen transformada y u y v sus coordenadas. Según esto la zona muerta valdrá $2(1\text{-}nz)\Delta_b$, con rango desde 0 a 4. El caso general se obtiene para $nz = 0$ con una zona muerta de 2.

Tomando u y v como coordenadas del espacio transformado wavelet se define para cada bloque i de cada subbanda b, con un ancho de cuantificación Δ_i que será el mismo para cada bloque de cada banda y $q_i(u,v)$ como coeficiente cuantizado.

El ancho de cuantificación viene dado por:

$$\Delta_b = 2^{R_b - e_b}\left(1 + \frac{\mu_b}{2^{11}}\right) \qquad (1.9)$$

Donde R_b son los bits que representan el rango dinámico de la sub-banda b. Añadiendo los bits de guarda G_0, se tiene que $R_b = R_I - G_0$. Además ϵ_b será el exponente y μ_b la mantisa. Así cuanto mayor sea el exponente, más pequeño será el ancho de cuantificación (figura 1.6). Si se asume que el número de bits necesarios para manejar la información vendrá dado por el rango dinámico de la imagen transformada dividido por el ancho de cuantificación según la siguiente relación se obtiene un valor directo del número de planos de bit.

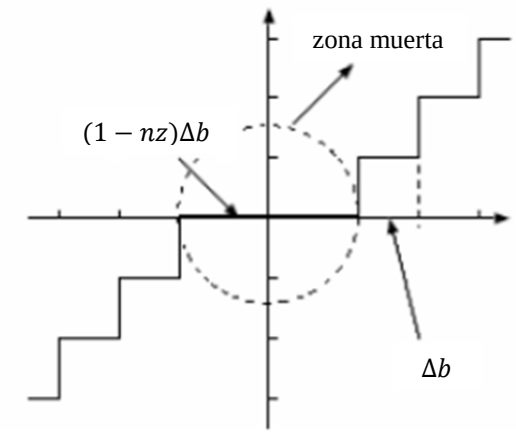


Figura 1.6 Cuantificador uniforme con zona muerta variable.

$$M_b = \left| log_2 \frac{2^{R_{I-1}}}{\Delta_b} \right| = G_0 + \epsilon_b - 1 \quad (1.10)$$

Los únicos parámetros que debe conocer el decodificador son el exponente y la mantisa.
Para ello hay dos maneras: con el valor (ϵ_b, μ_b) para cada subbanda, o bien con el valor (ϵ_0, μ_0) de la banda LL y el resto se obtiene mediante la relación:

$$(\epsilon_b, \mu_b) = (\epsilon_0 - N_L + n_b, \mu_0) \quad (1.11)$$

Donde N_L es el nivel de descomposición y n_b la subbanda.

El proceso de reconstrucción es inmediato. Sólo queda añadir que la reconstrucción de las muestras se hace en el medio del intervalo, aunque también existen reconstrucciones a 37.5% del intervalo.
Como es lógico, la cuantificación sólo tendrá sentido cuando se utilicen filtros no reversibles (9,7), o lo que es lo mismo, para filtros reversibles (5,3) el ancho de cuantificación será $\Delta_b = 1$ con $\epsilon_b = R_b$ y $\mu_b = 0$.

1.3.4 Codificación de entropía

Una vez pre-procesada, transformada y cuantificada la imagen se pasa a la etapa de codificación propiamente dicha, donde tiene lugar la compresión y se forma la trama de bits. La codificación de entropía ya se utilizaba en anteriores y muy conocidos algoritmos de compresión, como SPIHT [9] o EZW [8], que aprovechaban la correlación entre las distintas bandas de la transformada wavelet. Si se explota esta propiedad de la transformada, puesto que no es más que redundancia, se limita la flexibilidad de ordenación de la trama y además se disminuye la resistencia al error, puesto que puede haber propagación de errores. Estas son dos características importantes en el estándar JPEG2000 para aplicaciones de transmisión de datos. Evidentemente los algoritmos anteriores son muy potentes si lo único que se pretende es almacenar y no transmitir. Por tanto el JPEG2000 tratará cada subbanda por independiente. El algoritmo de compresión, otro de los núcleos del estándar junto a la transformada wavelet, es el EBCOT (Embebbed Block Coding with Optimized Truncation)[10]. El algoritmo EBCOT consiste en fragmentar la distintas subbandas en

bloques de tamaño arbitrario y realizar una codificación por planos de bit. Como se mencionó antes la codificación de bloques independientes tiene como ventajas el poder tener un acceso aleatorio a la imagen, el poder codificar en paralelo reduciendo el tiempo de cálculo, poder cortar y rotar la imagen una vez procesada, control de tasas, variación del orden e introducir códigos de resistencia al error. Por lo contrario, aunque se aumenta la flexibilidad y con ello el número de posibles aplicaciones, la eficiencia de compresión va a ser menor debido a que no se explota dicha correlación entre bandas, pero es asumible.

1.4 Conclusiones

El uso masivo de internet durante los últimos años, la proliferación de los dispositivos multimedia en el ámbito cotidiano, empresarial y personal, han llevado a nuevas y diferentes demandas por parte de los usuarios en tan diversas áreas como lo son aplicaciones médicas, bibliotecas digitales, entretenimiento, etc., lo que requiere métodos de acceso y gestión de la administración mucho más eficientes y con rápida capacidad de respuesta. Con éste capítulo logramos dar una amplia perspectiva respecto hacia dónde va enfocada nuestra investigación, centrándose en la cuestión ¿Qué características resultan más precisas y descriptivas en cuanto a la representación del contenido en un video?. En la actualidad existe un enorme interés por desarrollar descriptores audiovisuales que permitan caracterizar el contenido de las imágenes en forma automatizada. Valiéndose del fundamento MPEG7 en combinación con el principio de la Transformada de Wavelet, conseguimos planificar el diseño un método capaz de clasificar y agrupar las escenas principales dentro de una secuencia de video.

Capítulo 2

Aplicación del estándar JPEG2000-representación de los datos

Uno de los problemas con los que se cuenta al trabajar con imágenes digitales, es que requieren de un gran poder de procesamiento, de mucho espacio en memoria para ser almacenadas y de mucho ancho de banda para ser transmitidas. Pensando en ello y con el objetivo de trabajar con menor información al obtener cada uno de los frames de video y al mismo tiempo consiguiendo que dicha información sea valiosa y relevante, se propone trabajar bajo el estándar JPEG2000, puesto que éste cuenta con las características de: efectuar compresión con pérdidas o sin pérdidas, a una compresión son posibles múltiples descompresiones, ofrece decodificación progresiva, escalabilidad en resolución y calidad (escalabilidad en SNR), codificación de regiones de interés y resistencia a errores. JPEG2000 [6] considera individualmente cada trama de video, el cual permite la compresión de diferentes tipos de imágenes (como lo son imágenes binarias, a nivel de gris, hiper-espectrales, entre otras) con características diferentes (naturales, científicas, de teledetección, gráficas, compuestas) y de gran tamaño (superiores a los 64K x 64K píxeles). JPEG200 cuenta con un rendimiento superior de compresión, cuyas ganancias de compresión se atribuyen al uso de la transformada wavelet discreta y a un sistema más sofisticado de codificación de la entropía. También permite la representación de resolución múltiple, la cual puede emplearse para otros propósitos más allá de la compresión de dicha imagen. Así mismo tiene la característica de efectuar una transmisión progresiva de precisión por píxel y resolución, comúnmente llamada decodificación progresiva. Con JPEG2000 es capaz de obtener compresiones sin pérdidas y con pérdidas, en donde la compresión sin pérdida en JPEG2000 se basa en el uso de una transformada wavelet entero a entero reversible [6]. Este estándar

cuenta con la ventaja de permitir el acceso y procesamiento aleatorio del flujo de código, referido a la región de interés (ROI), de esta manera es posible almacenar diferentes partes de la misma imagen con diferente calidad.

Teniendo en cuenta el objetivo de éste trabajo de tesis, se adquiere ventaja al hacer uso del estándar JPEG2000, ya que en una primera etapa, gracias a la Transformada Discreta de Wavelet, se obtienen las imágenes compresas (subbanda LL) respecto a las imágenes originales de cada frame de video, además de al mismo tiempo contar con las imágenes que proporcionan información de bordes en cada una de las subbandas generadas por la Transformada Discreta de Wavelet (LH, HL y HH). Así bien, se propone la utilización de compresión con pérdidas por medio de la Transformada de Wavelet, Daubechies 9/7, puesto que las imágenes compresas empleadas no requieren una réplica exacta respecto a la original, (no como aquellas imágenes médicas o satelitales con un fin de estudio o diagnóstico estrictamente exacto), haciendo uso de los coeficientes en baja frecuencia asociados a la representación de flujo de ondas, obtenidos mediante filtrado. Para ello son utilizados dos tipos de filtros, el filtro paso bajas y el filtro paso altas, para con ello poder realizar la Transformación Discreta de Ondas, ondas dinámicas centradas en cero.

2.1 Preparación de los datos

Esta etapa tiene por objetivo alistar los datos para la aplicación de la Transformada Discreta de Wavelet.

Como primera fase, se realiza una transformación de color irreversible a cada una de las imágenes, puesto que los datos de entrada de las imágenes se encuentran en formato RGB. Lo que permite obtención de las imágenes en componentes YC_rC_b, separando la luminancia de la crominancia. Dicha transformación está dada por la expresión (1.3). *Ver sección 1.3.1.1*.

$$\begin{pmatrix} Y \\ C_r \\ C_b \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.16875 & -0.33126 & 0.5 \\ 0.5 & -0.41869 & -0.08131 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \qquad (1.3)$$

2.2 La Transformada de Wavelet Discreta

Ahora bien, efectuar el análisis de ondas en función de una imagen conlleva un análisis multiresolución. Mallat [14] propone dicho cálculo haciendo uso de bancos de filtros. Los bancos de filtros utilizados aquí son asociados dos filtros por una dimensión: un filtro paso bajas (La) y un filtro paso altas (Ha). Recordando que filtrar un vector unidimensional (en este caso cada fila y columna de la imagen) resulta de la convolución de ese vector con la respuesta impulso discreta y filtros finitos. Así pues la convolución Î, de una línea I_l con el filtro paso bajas H de longitud T, está dada por:

$$\hat{I}_l = I_l * H \quad ; \quad \hat{I}_l(t) = \sum_{-T/2}^{T/2} I_l(t-\tau)H(\tau) \quad (2.1)$$

En primer lugar las líneas de la imagen son procesadas por el filtro paso-bajas y el filtro paso-altas. Para obtener los coeficientes de transformación de ondas, estos filtros son aplicados separadamente a las filas y a las columnas de la imagen. Esto da como resultado un número de coeficientes 2 veces más que en la imagen original, para lo que la imagen de cada filtro es submuestreada(↓) por un factor de 2. Las columnas son filtradas (paso bajo- paso alto) y submuestreadas a continuación. Esta combinación de filtrado da como resultado 4 imágenes o sub bandas denotadas como LL1, LH1, HL1 y HH1 que forman el primer nivel de descomposición Wavelet.

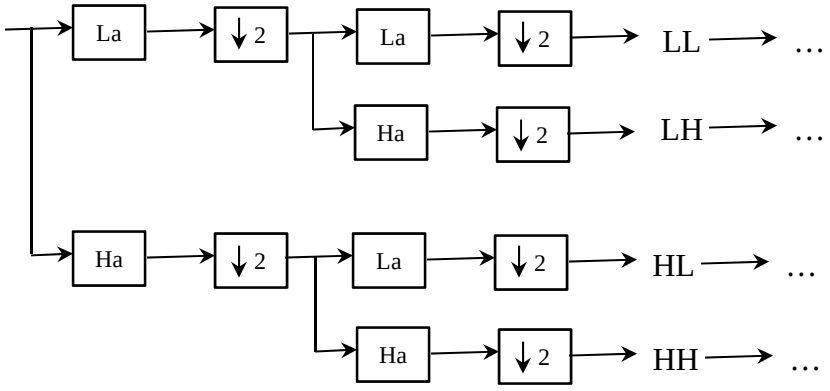


Figura 2 Análisis de la Transformada Discreta de Wavelet

Típicamente éste proceso es repetido K veces, resultando así $W_t = \{LL^0, LL^k, LH^k, HL^k, HH^k, k = 1 \dots K\}$, dónde k es el nivel de descomposición. W_t es la trama de onda en el instante t correspondiente a la imagen I_t en un instante t del video.

Cuando k=0 la imagen es la resolución completa, entendiendo como la subbanda LL^0, y cuando k=K se tiene la resolución mínima de la imagen.

Filtro de Análisis		
i	Filtro paso-bajas L_a	Filtro paso-altas H_a
0	0.6029490182363579	1.115087052456994
$\pm$1	0.2668641184428723	-0.591271763114247
$\pm$2	-0.07822326652898785	-0.05754352622849957
$\pm$3	-0.01686411844287495	0.09127176311424948
$\pm$4	0.02674875741080976	

TAB. 2 Coeficientes del Filtro de Ondas 9/7 de Deubechies

La sub-banda LL:

- Contiene una imagen reducida respecto a la imagen normal

La sub-banda HF

- Contiene la información de contorno
- LH: contornos horizontales
- HL: contornos verticales
- HH: contornos diagonales

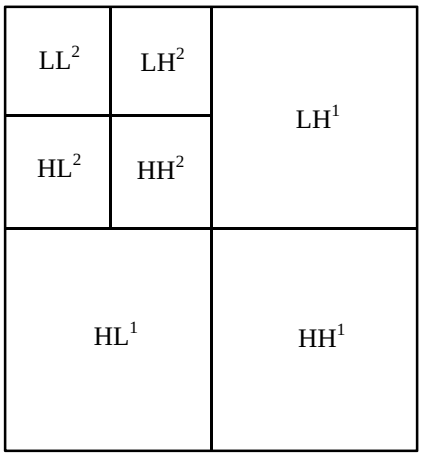


Figura 2.1 Notación de la descomposición mediante la Transformada Discreta de Wavelet

Componente Y

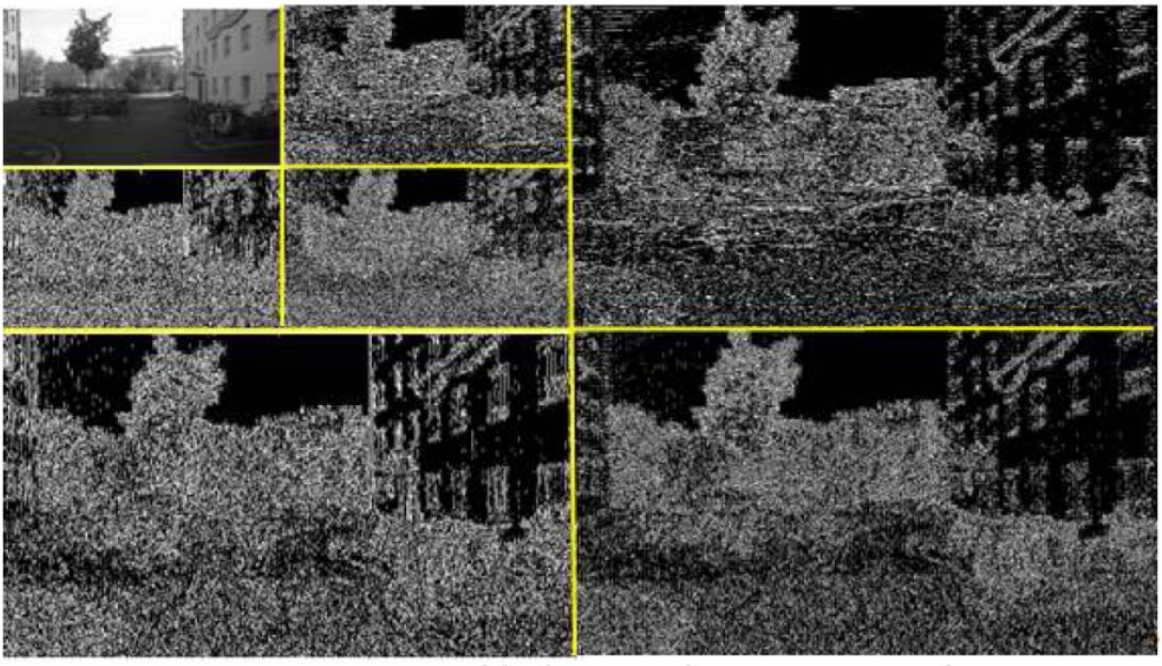


Figura 2.2 Descomposición Wavelet de una imagen en la componente Y, con K=3.

Componente C_r

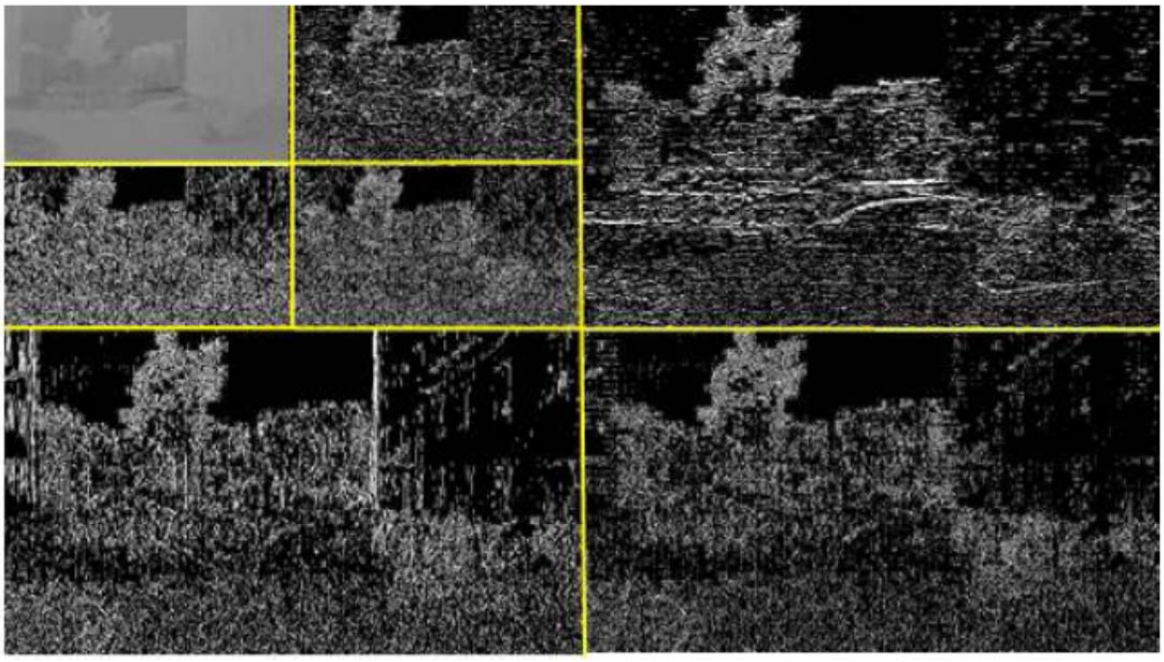


Figura 2.3 Descomposición Wavelet de una imagen en la componente C_r, con K=3.

Componente C_b

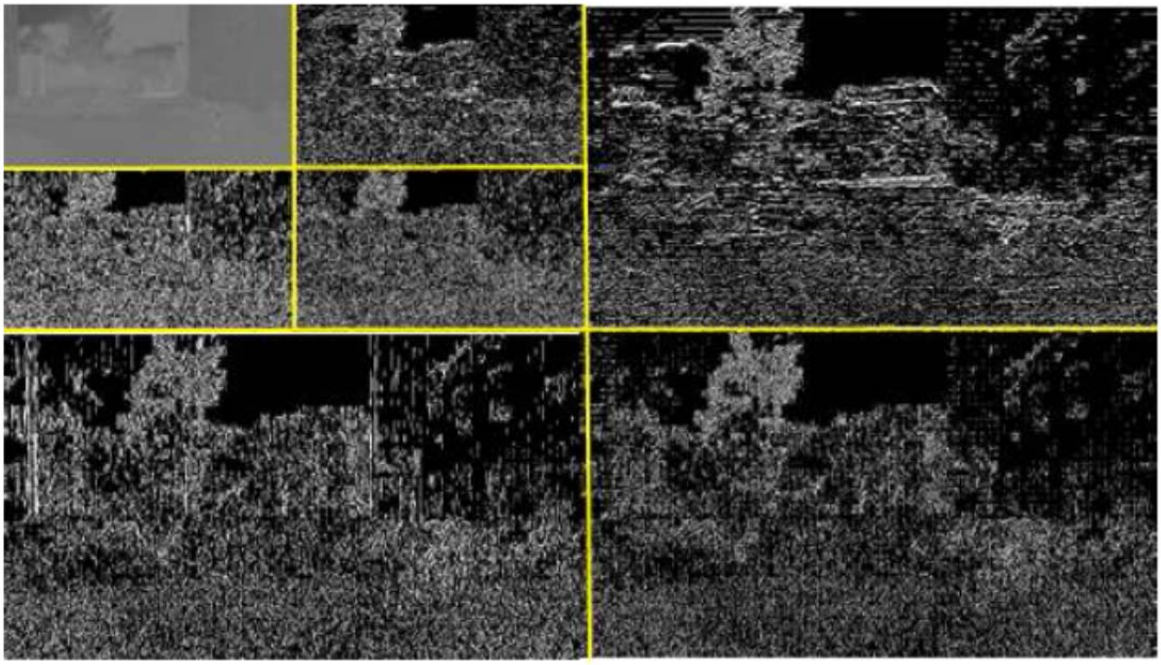


Figura 2.4 Descomposición Wavelet de una imagen en la componente C_b, con K=3.

2.3 Cálculo del Histograma

La indexación automática de videos por contenido, consiste en la definición y extracción automática de una o más características del video, así como también la definición de una medida de similitud entre estos descriptores. Estas características, son o bien en el dominio espacio-temporal o, simplemente, el video es considerado como un simple conjunto de imágenes.

2.3.1 Histograma de Color

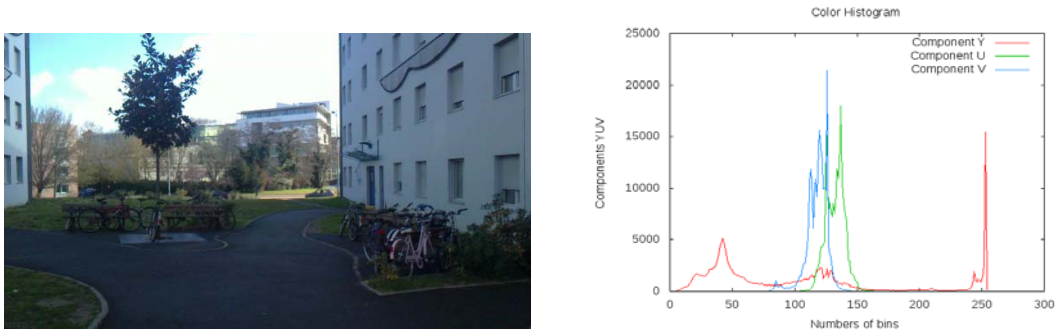
Para comparar imágenes, es necesario extraer una firma basada en sus píxeles y definir una regla para compararlas. La firma puede ser el color, la textura, la forma o cualquier otra información que permita llevar a cabo la comparación. Una de estas primeras formas que propone la literatura es el cálculo del Histograma de color [18]. El histograma de color representa la frecuencia de aparición de cada una de las intensidades de color presentes en la imagen, mediante la contabilidad de los pixeles que comparten dichos valores de intensidad

de color. El histograma está compuesto por diferentes rangos o contenedores que representan un valor o conjuntos de valores de intensidad de color. Anterior a la etapa de contabilización de cada uno de los valores de los pixeles, existe una etapa de cuantificación de los intervalos o contenedores que se refiere al proceso de reducción del número de intervalos agrupando colores cuyos valores están próximos entre sí en el mismo contenedor. Esta etapa es importante en cuanto a que la cuantificación de los intervalos reduce la información representada por el descriptor sobre la imagen al mismo tiempo que reduce el tiempo de cálculo.

Para ello se define un espacio de color dado por la cuantificación de color en una región y la frecuencia de ocurrencia de cada color cuantificado de la imagen. Así bien, el histograma de una imagen I se describe como:

$$H(I) = \{(C^m, card(C^m)), m = 1 \dots M\} \quad (2.2)$$

Dónde M es el número de colores de cuantificación y $C^m = (C_1^m, C_2^m, C_3^m)$ es el vector de color cuantificado dentro del espacio de representación considerado (por ejemplo RGB, YUV, HSV, …).



a) Frame obtenido del video b) Histograma a color YUV

Figura 2.5 Representación del histograma a color YUV

Para el trabajo aquí presentado el histograma a color es calculado directamente en el dominio de la Transformada Wavelet Discreta. Para esto se construye un histograma de baja frecuencia y otro de alta frecuencia. El histograma de baja frecuencia está representado por la subbanda LL y el histograma de lata frecuencia se encuentra representado por la suma del valor absoluto de las subbandas en H. Por lo que el histograma de alta frecuencia se define como:

$$I_{h_H} = \sum_{x=0}^{nbPixel} (Iw_{LH})_x + (Iw_{HL})_x + (Iw_{HH})_x \quad (2.3)$$

Donde I_{h_H}es el histograma de alta frecuencia de la imagen, $nbPixel$ es el número total de pixeles en la imagen wavelet, Iw_{LH} es la imagen wavelet en la subbanda LH, Iw_{HL} es la imagen wavelet en la subbanda HL, Iw_{HH} es la imagen wavelet en la subbanda HH. Tal que la subbanda H es agrupada, el histograma es calculado tal cual ha sido previamente descrito antes [19].

2.4 Conclusiones

El presente capítulo indica los cálculos trabajados en el dominio de la caracterización de cada imagen, es decir, en los principios fundamentales que consiguen describir cada frame perteneciente a un video. Se propone un método bajo el estándar JPEG2000, así como el uso de un histograma en baja y alta frecuencia, principal descriptor de las imágenes en análisis.

Capítulo 3

Clasificación de imágenes

El Reconocimiento de Formas constituye un amplio conjunto de técnicas para el tratamiento de datos entre las que se puede mencionar: la selección y extracción de características, la clasificación de un objeto en un grupo dado y la división de los datos en grupos (agrupamiento). Uno de los enfoques en el Reconocimiento de Formas, en función del tipo de espacio de representación utilizado es el Reconocimiento Estadístico de Formas que se apoya en la teoría de decisión, éste asume que el espacio de representación tiene una estructura de espacio vectorial y/o métrico y no se supone ninguna relación estructural entre las distintas características. El problema del agrupamiento puede definirse de la siguiente manera: dados n puntos en un espacio n-dimensional particionar los mismos en k grupos tales que los puntos dentro de un grupo son más similares que cada uno a los de los otros grupos, dicha similaridad se mide atendiendo a alguna función distancia (función de disimilardad) o alguna función de similaridad.

En el presente trabajo se asume que el video debe ser segmentado dentro de shots los cuales son mayormente representados por un conjunto de keyframes. Un shot es definido como una secuencia de frames que representan una acción continua en tiempo y espacio, que generalmente consiste en múltiples frames muchos de los cuales son muy similares en contenido, por lo que se desea representar cada shot por un mínimo conjunto de keyframes que capturen el contenido semántico del shot. Por lo que de ésta forma el video clustering es reducido a un agrupamiento keyframe de imágenes fijas. Una vez que las tomas han sido extraídas del video, es utilizada una técnica de clustering tipo jerárquico considerando un kernel de tipo Chi square para organizar dichos shots y de esta forma se consigan eficientes indexados, búsquedas o vistas.

3.1 Clustering Jerárquico

Los llamados métodos jerárquicos tienen por objetivo agrupar clusters para formar uno nuevo o bien, separar alguno ya existente para dar origen a otros dos, de tal forma que, si sucesivamente se va efectuando éste proceso de aglomeración o división, se minimice alguna distancia o bien se maximice alguna medida de similitud.

Los métodos jerárquicos se subdividen en aglomerativos y disociativos. Cada una de estas categorías presenta una gran diversidad de variantes [20].

1. Los métodos aglomerativos, también conocidos como ascendentes, comienzan el análisis con tantos grupos como individuos haya. A partir de estas unidades iniciales se van formando grupos, de forma ascendente, hasta que al final del proceso todos los casos tratados están englobados en un mismo conglomerado.
2. Los métodos disociativos, también llamados descendentes, constituyen el proceso inverso al anterior.

Comienzan con un conglomerado que engloba a todos los casos tratados y, a partir de este grupo inicial, a través de sucesivas divisiones, se van formando grupos cada vez más pequeños. Al final del proceso se tienen tantas agrupaciones como casos han sido tratados.
Para fijar ideas, centrémonos en los métodos aglomerativos. Sea n el conjunto de individuos de la muestra, de donde resulta el nivel $K = 0$, con n grupos. En el siguiente nivel se agruparán aquellos dos individuos que tengan la mayor similitud (o menor distancia), resultando así $n - 1$ grupos; a continuación, y siguiendo con la misma estrategia, se agruparán en el nivel posterior, aquellos dos individuos (o clusters ya formados) con menor distancia o mayor similitud; de esta forma, en el nivel L tendremos $n - L$ grupos formados. Si se continúa agrupando de esta forma, se llega al nivel $L = n - 1$en el que sólo hay un grupo, formado por todos los individuos de la muestra.
Esta manera de formar nuevos grupos tiene la particularidad de que si en un determinado nivel se agrupan dos clusters, estos quedan ya jerárquicamente agrupados para el resto de los niveles.
Los métodos jerárquicos permiten la construcción de un árbol de clasificación, que recibe el nombre dedendrograma (figura 3), en el cual se puede seguir de forma gráfica el procedimiento de unión seguido, mostrando que grupos se van uniendo, en qué nivel concreto

lo hacen, así como el valor de la medida de asociación entre los grupos cuando estos se agrupan.

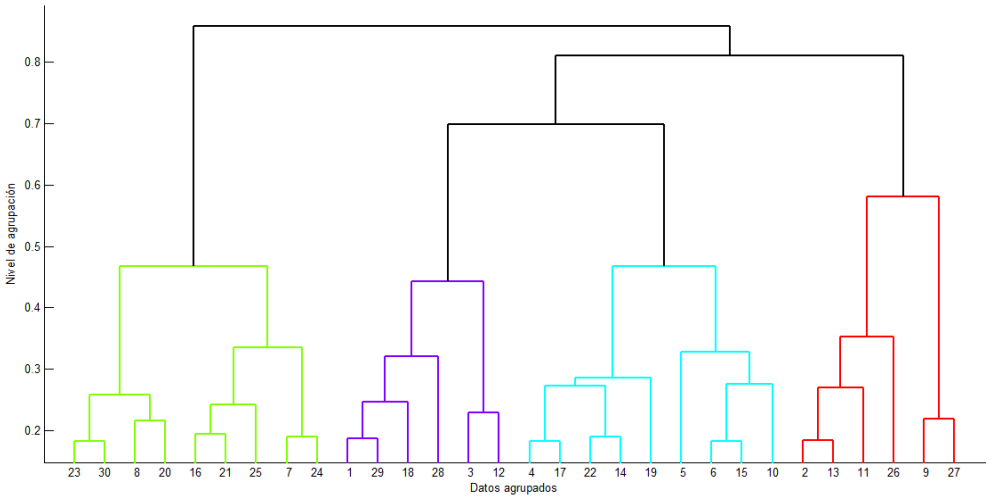


Figura 3. Ejemplo de dendrograma de Clustering Jerárquico Aglomerativo[2]

En resumen, la forma general de operar de estos métodos es bastante simple. Por ejemplo, en los métodos aglomerativos se parte de tantos grupos como individuos haya. A continuación se selecciona una medida disimilitud, agrupándose los dos grupos o clusters con mayor similitud. Así se continúa hasta que:

1. Se forma un solo grupo.
2. Se alcanza el número de grupos prefijado.
3. Se detecta, a través de un contraste de significación, que hay razones estadísticas para no continuar agrupando clusters, ya que los más similares no son lo suficientemente homogéneos como para determinar una misma agrupación.

Para nuestro trabajo, Zhang [27,28] propone un agrupamiento jerárquico de keyframes basado en múltiples características de la imagen para mejorar la eficiencia de la búsqueda. Los algoritmos basados en el agrupamiento jerárquico proporcionan como resultado una secuencia anidada de grupos de pixeles, que se puede representar en forma de árbol. Estos métodos se pueden llevar a cabo de dos formas distintas, uniendo pequeños clusters para formar otros mayores, o bien dividiendo clusters grandes en otros más pequeños.

[2] Ejemplo ejecutado del demo Images Processing Toolbox Matlab R2010- Hierarchical Clustering

La regla establecida para determinar cuán parecidas son las imágenes es definida con el cálculo de la distancia entre histogramas. Para ello se propone hacer uso de la función Chi-Square [22] definida por:

$$x^2\left(h_{i_p}, h_{j_p}\right) = 1 - \sum_{n=1}^{N} \frac{\left(h_{i_p}(n) - h_{j_p}(n)\right)^2}{\frac{1}{2}\left(h_{i_p}(n) + h_{j_p}(n)\right)} \tag{3.1}$$

En dónde h_{i_p} es el histograma observado, h_{j_p} el histograma a comparar, N el número total de bins.

Haciendo el análisis de Chi-square tomando en cuenta los canales YUV en la imagen:

$$x^2(h_i, h_j) = \frac{1}{3}\left(x^2\left(h_{i_Y}, h_{j_Y}\right) + x^2\left(h_{i_U}, h_{j_U}\right) + x^2\left(h_{i_V}, h_{j_V}\right)\right) \tag{3.2}$$

De (3.1) una distancia reducida es definida en el kernel x^2 [22]:

Si consideramos a X como el nuevo espacio en el que son extraídas características visuales, es definida una función kernel k sobre XxX como una función que evalúa la similaridad, la correlación, entre dos descriptores de datos. Una definición de kernel es $>$ si podemos encontrar una función embebida $\emptyset$ *(injection)* la cual mapea cualquier dato x hacia un vector $\emptyset(x)$ en el espacio de Hilbert (espacio inducido o espacio de características), entonces la función k por el siguiente producto en el espacio inducido:

$$K(X, Y) = \langle \varphi(X), \varphi(Y) \rangle \tag{3.3}$$

Es una función kernel sobre X. La función embebida $\emptyset$, puede estar definida implícita o explícitamente. Esta definición también proporciona dos formas de construir una función kernel k ya sea mapeándola $\emptyset$ ó a partir de una función kernel ya valida k' y modificándola función de mapeo $\emptyset$', con respecto a las propiedades bilineales del producto punto, con el fin de diseñar el mapeo final en la función $\emptyset$ de k.

Esta definición es también la razón de que haya un gran interés por funciones del kernel: como un producto de punto (en un espacio inducido), que idealmente reemplaza productos punto en la toma de decisión en algoritmos (como redes de regresión, clasificación/neuronal, máquinas de soporte vectorial, entre otras) que linealmente estiman una función de decisión que proveen estos algoritmos con la capacidad de estimación decisión no lineal [22].

Y definiendo la distancia como:

$$d = \left\|\varphi(h_i) - \varphi\left(h_j\right)\right\|^2 \tag{3.4}$$

$$x^2(h_i, h_i) = 1 x^2\left(h_j, h_j\right) = 1$$

$$\langle\left(\varphi(h_i) - \varphi\left(h_j\right)\right) - \left(\varphi(h_i) - \varphi\left(h_j\right)\right)\rangle = \langle\varphi(h_i), \varphi(h_i)\rangle - 2\langle\varphi(h_i), \varphi\left(h_j\right)\rangle + \langle\varphi(h_j), \varphi\left(h_j\right)\rangle =$$

$$2 - 2\langle\varphi(h_i), \varphi\left(h_j\right)\rangle$$

$$d = 2 - 2x^2\left(h_i, h_j\right) \tag{3.5}$$

Para finalmente conseguimos con d obtener la medida de similitud entre histogramas. La idea original del cálculo de dicha distancia reside en que a través de ella se consiga alinear secuencias. Esta medida indica el número de transformaciones necesarias para transformar un bloque en otro. Tal como lo propone Smith-Waterman en su algoritmo bio-informático [25]. Para ello es importante recordar que contaremos con un número de distancias calculadas igual a:

$$n\acute{u}m_d = \frac{Tot_{frames} * \left(Tot_{frames} - 1\right)}{2} \tag{3.6}$$

En donde $n\acute{u}m_d$ número de distancias calculadas en un video, Tot_{frames} número total de frames contenidos en el mismo vídeo. Dicha fórmula radica en una matriz triangular superior, en la cual resulta innecesario contar con el cálculo de los elementos pertenecientes a diagonal principal, pues estos corresponden al cálculo de la distancia en dónde el elemento i y el elmento $j$son iguales, es decir, se estaría comparando un frame con el mismo frame.

La figura 3.1 muestra de una forma visual, un ejemplo haciendo uso de la medida de disimilitud planteada en la ecuación 3.5, a partir del cálculo del histograma en alta frecuencia (ec. 2.3). Contemplando un video de474 frames. En dicho ejemplo se muestran algunas de las distancias calculadas, en dónde la distancia 1 corresponde al frame 1 y el frame 2, la distancia 2 hace referencia a aquella entre el frame 1 y el frame 3 del video, así sucesivamente hasta llegar a la distancia número 112101, la cual corresponde a aquella entre el frame 473 y el frame 474, cumpliendo así con los cálculos de la forma triangular superior.

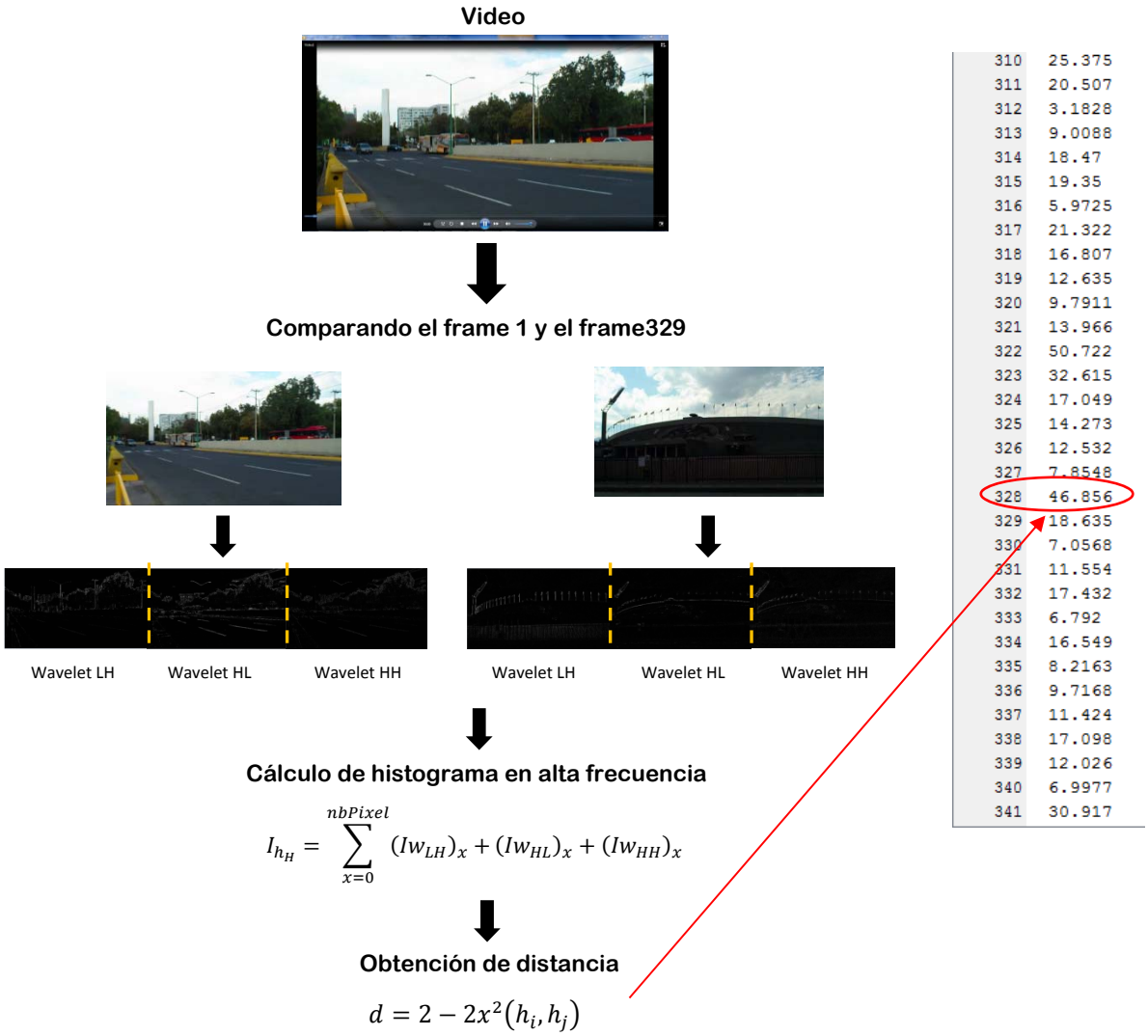


Figura 3.1 Ejemplo visual de la aplicación de disimilitud de la ecuación 3.5

Una vez elegida la medida de disimilitud que se vaya a emplear, se pasa a elegir el algoritmo de clustering. Los resultados que se obtienen con cada algoritmo pueden ser distintos, a pesar de que la imagen de partida y la medida de disimilitud empleadas sean las mismas. Además, diferentes valores de k (número de clusters) suelen generar diferentes tipos de clusters. La inicialización de los valores de los centroides de los clusters también es un hecho crítico, ya que algunos clusters podrían quedar vacíos si el centroide inicial queda lejos de la distribución de los objetos. El algoritmo k-means es uno de los criterios de agrupamiento más ampliamente utilizado y se basa en la minimización de un índice de prestaciones, el cual se define como la suma de las distancias al cuadrado de todos los puntos u objetos incluidos en un cluster al centroide de dicho cluster. El número de clusters k es definido a priori. El

comportamiento de este algoritmo depende del número de clusters especificado, la elección inicial de los centroides y el número de objetos en la imagen [26].En nuestro trabajo decidimos hacer uso de un algoritmo de clustering jerárquico el cual puede es derivado de los que se describirán en los siguientes subtemas.

3.1.1 Single link

En este método la distancia entre dos clusters es calculada como la distancia entre los dos elementos más cercanos en ambos clusters. La distancia D(X,Y) entre el cluster X y Y se representada por la expresión [20]:

$$D(X,Y) = \min_{x \in X, y \in Y} d(x,y) \qquad (3.7)$$

Donde X y Y son dos conjuntos de elementos considerados como clusters y d(*x,y*) denota la distancia entre dos elementos de x y y.

3.1.2 Mean link

En el enlace medio de agrupamiento, se considera que la distancia entre un grupo y otro grupo debe ser igual a la distancia media desde cualquier miembro de un grupo a cualquier miembro del otro grupo. Por lo tanto, la distancia media D(X, Y) entre cada objeto que pertenece a grupos X e Y se definen como [21]:

$$D(X,Y) = \frac{1}{N_X \times N_Y} \sum_{i=1}^{N_X} \sum_{j=1}^{N_Y} d(x_i, y_j) \; ; x_i \in X, y_j \in Y \qquad (3.8)$$

Donde:

$d(x,y)$ es la distancia media entre los objetos $x \in X$y $y \in Y$;

X y Y son los dos conjuntos de objetos(clusters);

N_X y N_Y son el número de objetos en el cluster$X$ y Y respectivamente.

3.1.3 Complete link

En el enlace completo las distancias entre dos clusters son calculadas como la máxima de distancias entre un par de objetos, un objeto de ellos perteneciente a un cluster y otro a otro cluster respecto al anterior. La distancia D(X,Y) entre el cluster X y Y está dado por la expresión [21]:

$$D(X,Y) = \max_{x \in X, y \in Y} d(x,y) \tag{3.9}$$

Donde X y Y son dos conjuntos de elementos considerados como clusters y $d(x,y)$denota la distancia entre dos elementos de x y y.

En la figura 3.3, se aprecia el dendrograma de un ejemplo de un clustering jerárquico aglomerativo aplicado a un video de 473 frames, utilizando como descriptor un histograma de baja frecuencia (ec. 2.2) a partir del cálculo de los coeficientes Wavelets en la subbanda LL con un nivel de descomposición k=1 (TAB. 2.1). Se utiliza la medida de disimilitud planteada en la ecuación 3.5, para finalmente con ello utilizar un enlace de tipo mean link (ec. 3.7), tomando en cuenta 7 centros de cluster, es decir, obteniendo únicamente las 7 imágenes clave del video.

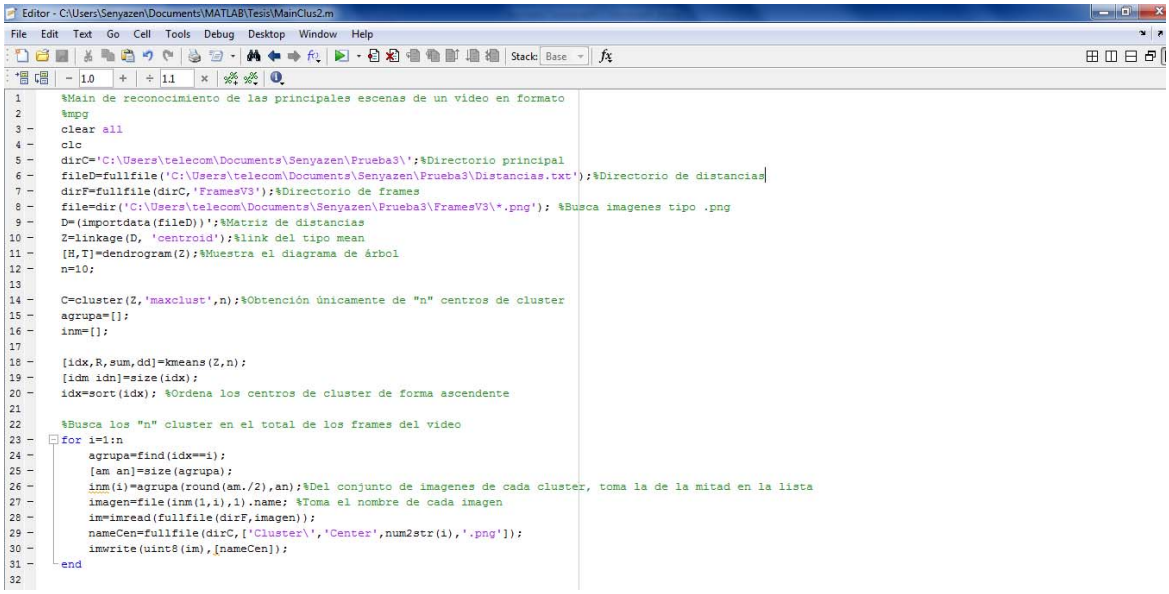


Figura 3.2 Ejemplo de la programación principal en Matlab para la obtención de las escenas principales en un video al aplicar clustering jerárquico

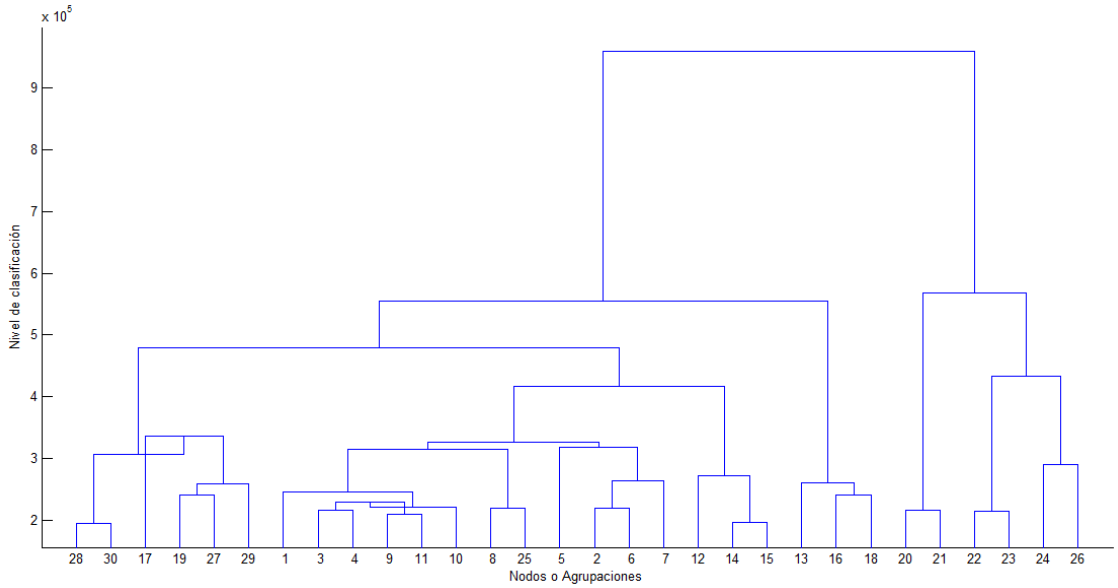
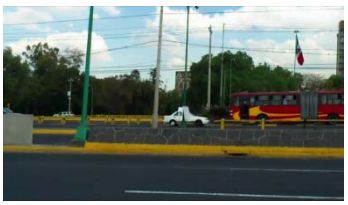


Figura 3.3 Dendrograma correspondiente a la aplicación de clustering jerárquico a un video de 373 frames

La TAB. 2.2 contiene los resultados obtenidos a partir del clustering jerárquico de la figura 3.4 para 7 centros de cluster.

Número de Cluster	**Número de frames agrupados**	**Centro de Cluster**	**Imagen (centro de cluster)**
1	164	83	
2	9	169	

3	129	238	
4	3	304	
5	96	354	
6	25	414	
7	46	450	

TAB 3 Resultados de clustering jerárquico aglomerativo de un video de 373 frames.

3.2 Máquinas de Soporte Vectorial (SVM's)

La teoría de las SVM's fue desarrollada inicialmente por V. Vapnik [29] a principios de los años 80 y se centra en lo que se conoce como Teoría del Aprendizaje Estadístico. El objeto de las SVM's es dar solución al problema fundamental que surge en distintos campos, donde se estudia ,la relación entre sesgo y varianza [30], el control de la capacidad [31], sobreajuste en los datos [32], etc. Este problema consiste en buscar, para una tarea de aprendizaje dada, con una cantidad finita de datos, una adecuada función que permita llevar a cabo una buena generalización que sea resultado de una adecuada relación entre la precisión alcanzada con un particular conjunto de entrenamiento y la capacidad del modelo.

El objetivo principal de este tipo de algoritmos es aprender a partir de los datos y para ello busca la existencia de alguna dependencia fundamental entre un conjunto de vectores de entrada

$$\{x_i, i = 1, \dots, n\} \subseteq \mathcal{X} \subseteq \mathbb{R}^d \tag{3.9}$$

Y de valores salida

$$\{y_i, i = 1, \dots, n\} \subseteq \mathcal{Y} \subseteq \mathbb{R} \tag{3.10}$$

El modelo representado por la figura 3.4 (denominado modelo de aprendizaje a partir de ejemplos) recoge de manera clara el objetivo que se persigue. En este esquema, G representa un modelo generador de datos que nos proporciona los vectores $x_i \in \mathcal{X}$, independientes e idénticamente distribuidos de acuerdo con una función de distribución $F_{\mathcal{X}}(x)$desconocida pero que suponemos no varía a lo largo del proceso de aprendizaje. Cada vector x_i es la entrada del operador objetivo S, el cual lo transforma en un valor y_i según una función de distribución condicional $F_{\mathcal{Y}/\mathcal{X}=x_i}(y)$. Así la máquina de aprendizaje, que denotamos LM (Learning Machine) recoge el siguiente conjunto de entrenamiento,

$$\mathcal{Z} = \{(x_1, y_1), \dots, (x_n, y_n)\} \subseteq \mathcal{X} \times \mathcal{Y} = \mathcal{Z} \tag{3.11}$$

el cual es obtenido independiente e idénticamente distribuido siguiendo la función de distribución conjunta:

$$F(\mathcal{X},\mathcal{Y})(x,y) = F_{\mathcal{X}}(x)F_{\mathcal{Y}}/\mathcal{X} = x(y) \tag{3.12}$$

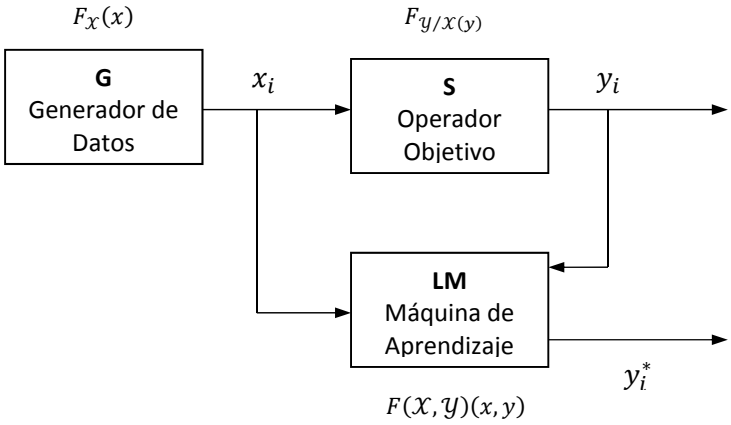


Figura 3.4 Esquema de configuración de una máquina de aprendizaje a partir de ejemplos

A partir del conjunto de entrenamiento Z, la máquina de aprendizaje construye una aproximación al operador desconocido la cual proporcione para un generador G, la mejor aproximación (en algún sentido) a las salidas proporcionadas por el supervisor. Formalmente construir un operador significa que la máquina de aprendizaje implementa un conjunto de funciones, de tal forma que durante el proceso de aprendizaje, elige una función apropiada siguiendo una determinada regla de decisión.

La estimación de ésta dependencia estocástica basada en un conjunto de datos trata de aproximar la función de distribución condicional $F_Y/X(y)$ lo cual en general lleva a un problema realmente complicado [29,33]. Sin embargo, el conocimiento de la función $F_Y/X(y)$ no siempre es necesario, a menudo se está interesado sólo en alguna de sus características. Por ejemplo, se puede buscar estimar la función de esperanza matemática condicional:

$$E[Y/X = x] = y\, dF_Y/X(y) \tag{3.13}$$

Por ello el objetivo del problema es la construcción de una función $F(X,Y)$ dentro de una determinada clase de funciones F elegida apriori, la cual debe cumplir un determinado criterio de la mejor manera posible. Formalmente el problema se plantea:

Dado un subespacio vectorial Z de $\mathbb{R}^{d+1}$ donde se tiene definida una medida de probabilidad $F_Z(z)$, un conjunto $F = \{f(z), z \in Z\}$, de funciones reales y función al $R: F \to \mathbb{R}$

Buscar una función $f^* \in F$ tal que

$$R[f^*] = \underset{f \in F}{mín} R[f] \tag{3.14}$$

Con el objeto de ser lo más general posible sería bueno elegir la función R de tal manera que se pudiese plantear con él el mayor número de problemas posibles. Por ello de define $R[f^*]$ como:

Definición 1.1 Dada una clase $F = \{f(z), z \in Z\}$ de funciones reales y una medida de probabilidad $F_Z(z)$ se define el riesgo, $R: F \to \mathbb{R}$, como:

$$R[f] = \int_Z c\big(z, f(z)\big)\, dF_Z(z) \tag{3.15}$$

donde $R[f]$ se denomina función de pérdida (o de coste) y toma valores no negativos.

A la vista de la figura 3, se llega a la conclusión de que los valores y_i e y^* coinciden cuando ésta máquina de aprendizaje habrá cometido un error que se debe cuantificar de alguna forma y éste es precisamente el sentido que tiene una función de pérdida.

Así en éste planteamiento, dado un conjunto $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, el principal problema consiste en formular un criterio constructivo para elegir una función de F puesto que el funcional (3.14) por sí mismo no sirve como criterio de selección, ya que la función $F_Z(z)$ incluida en él es desconocida. Para elaborar dicho criterio se debe tener en cuenta que el riesgo se define como la esperanza matemática de una variable aleatoria respecto a una medida de probabilidad, por tanto es lógico como estimación, la medida muestral y de aquí la siguiente definición:

> Definición 1.2 dado un riesgo definido por (1), un conjunto de funciones F y una muestra $\{z_1, \ldots, z_n\}$, al funcional $R_{emp}: F \to \mathbb{R}$ definido como
>
> $$R_{emp}[f] = \frac{1}{n}\sum_{i=1}^{n} c\big(z_i, f(z_i)\big), \quad f \in F \tag{3.16}$$
>
> se le domina riesgo empírico.

La forma clásica de abordar estos problemas es: sí el valor mínimo del riesgo se alcanzó con una función f_0 y el mínimo del riesgo empírico con f_n para una muestra dada de tamaño n ,

entonces se considera que f_n es una aproximación a f_0 en un determinado espacio métrico. El principio que resuelve éste problema se denomina principio de minimización de riesgo empírico. Este es el principio utilizado en los desarrollos clásicos, por ejemplo, cuando se plantea a partir de un conjunto de datos la regresión lineal mínimo cuadrática.

3.2.1 Modelos lineales de vectores de soporte

Denotamos las dos posibles etiquetas por $Y = \{-1, 1\}$ y

> Definición 2.1 Un conjunto de vectores $\{(x_1, y_1), \dots, (x_n, y_n)\}$donde $x_i \in \mathbb{R}^d$e$y_i \in \{-1,1\}$ para $i = 1, \dots, n$ se dice separable sí existe algún hiperplano en $\mathbb{R}^d$ que separa los vectores $X = \{x_1, \dots, x_n\}$ con etiqueta $y_i = 1$ de aquellos con etiqueta $y_i = -1$.

Dado un conjunto separable existe (al menos) un hiperplano

$$\pi: w * x + b = 0 \tag{3.17}$$

que separa los vectores $x_i, i = 1, \dots, n$ (figura 3.5) . Las SVM's buscan entre todos los hiperplanos, aquel separador que maximice la distancia de separación entre los conjuntos $\{x_i, 1\}$ y $\{x_i, -1\}$ (las dos clases posibles).

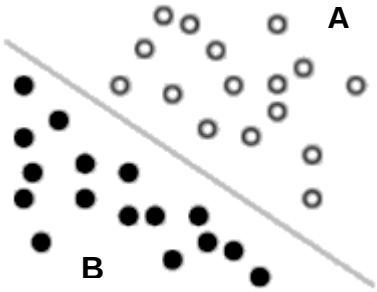


Figura 3.5 Conjunto e hiperplano separador en $\mathbb{R}^2$. Los puntos huecos representan los vectores con etiqueta $y = 1$ y los restantes con etiqueta $y = -1$.

De entre las posibles elecciones de los hiperplanos π_1 y π_2, parece natural elegir a aquella que proporcione una mayor separación entre ellos, ya que de ésta forma permitiría distinguir de forma más clara las regiones donde caen los puntos con distintas etiquetas. Planteándose

$$\min_{w \in \mathbb{R}^d} \frac{1}{2} \| w \|^2 \tag{3.18}$$

$$s.a. \quad y_i(x_i * w + b) - 1 \geq 0, \qquad i = 1, \dots, n$$

La solución para el caso de dimensión dos se puede interpretar gráficamente a partir de la figura 3.6. A la vista de ésta figura, es fácil darse cuenta de una característica muy importante de las SVM's y es que si se añade o elimina cualquier número de vectores que cumplan la desigualdad descrita, la solución del problema de optimización no se ve afectada. Sin embargo, basta con añadir un vector que se encuentre entre los dos hiperplano (vectores de soporte), para que la solución cambie totalmente.

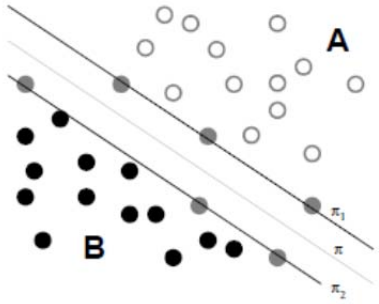


Figura 3.6 Hiperplanos paralelos y vectores soporte en $\mathbb{R}^2$.

Para resolver el problema de optimización con restricciones, se utiliza los multiplicadores de Lagrange. Así la función objetivo es:

$$L_p(w, b, \alpha_i) = \frac{1}{2} \| w \|^2 - \sum_{i=1}^{n} \alpha_i (y_i(x_i * w + b) - 1) \tag{3.19}$$

El problema queda como un problema de programación cuadrática donde la función objetivo es convexa, y los vectores que satisfacen las restricciones forman un conjunto convexo. Esto significa que se puede resolver el siguiente problema dual asociado al problema primal: maximizar la función $L_P(w, b, \alpha_i)$ respecto a las variables duales α_i sujetas a las restricciones impuestas para los gradientes L_P con respecto a w y b sean nulos, y sujeta también al conjunto de restricciones $C_2 = \{\alpha_i \geq 0, i = 1, \dots, n\}$. La solución de este problema se expresa en la forma:

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i, \quad \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{3.20}$$

y la función objetivo dual:

$$L_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i x_j \tag{3.21}$$

Los vectores del conjunto entrenamiento que proporcionan un multiplicador $\alpha_i > 0$ son denominados vectores de soporte y claramente, estos vectores se encuentran en uno de los hiperplanos π_1 ó π_2 . Para éste tipo de modelo de aprendizaje, los vectores soporte son elementos críticos ya que ellos son los que proporcionan la aproximación del problema, puesto que si todos los restantes elementos del conjunto de entrenamiento son eliminados (o son cambiados por otros que no se encuentran entre los dos hiperplanos), y se repite el problema de optimización, se encuentran los mismos hiperplanos separadores.

3.2.2 SVM's para datos no separables

En la práctica no es habitual trabajar con conjuntos separables. En estos casos (figura 3.7) se encuentran vectores de una clase dentro de la región correspondiente a los vectores de otra clase y por tanto nunca podrán ser separados de ésta clase por medio de hiperplanos.
En estas situaciones se diría que el conjunto es no separable. Ante estos casos, el problema de optimización (ec. 3.19) no encuentra una solución posible, lo que es evidente sí se observa como la función objetivo (ec. 3.19) crece de forma arbitraria y a que el multiplicador Lagrange correspondiente a este vector se puede tomar arbitrariamente grande sin que viole las restricciones. Sin embargo, no es difícil ampliar las ideas generales del caso separable al caso no separable introduciendo la variable ε de holgura en las restricciones y plantear un nuevo conjunto de restricciones:

$$x_i * w + b \geq +1 - \varepsilon_i \quad para\ y_i = +1$$
$$x_i * w + b \leq -1 + \varepsilon_i \quad para\ y_i = -1$$
$$\varepsilon_i \geq 0\ \forall i = 1, \dots, n \tag{3.22}$$

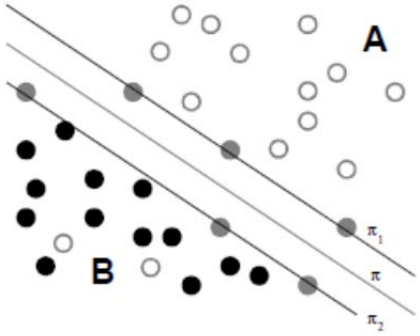


Figura 3.7 Ejemplo de hiperplanos separadores para el caso de datos no separables

Se tiene ahora que para que se produzca un error en la clasificación de un vector de entrenamiento (una entrada no ubicada en la clase correcta) es necesario que el valor correspondiente ε_i sea superior a la unidad. Así, si en el vector x_i se comete un error entonces $\varepsilon_i \geq 1$ y por lo tanto $i\varepsilon_i$ es una cota superior del número de errores que se cometen dentro del conjunto de entrenamiento.

Ya que en el caso no separable, necesariamente se han de cometer errores, parece natural asignar a la función objetivo un costo extra que en cierto modo penalice los errores (función de pérdida). Por todo ello, una opción lógica sería plantear el problema de minimizar.

$$\frac{1}{2} \| w \|^2 + C \sum_{i=1}^{n} \varepsilon_i^k \qquad (3.23)$$

Así bien, en la figura 3.8 podemos apreciar un ejemplo de aplicación de SVM's para un conjunto de datos correspondientes a 150 muestras de flores ornamentales (lirios), contando con 4 mediciones diferentes por cada una de ellas (p.e. longitud y anchura de pétalos y sépalos) las cuales determinan 3 especies de flor de lirio: *lirio setosa, lirio virginica, lirio versicolor*. Dichos datos son clasificados haciendo uso de un kernel lineal, agrupando en 2 conjuntos: aquellas flores que correspondan a la especie de flor del tipo *"setosa"* y aquellas que no pertenecen a dicha especie. Obteniendo éste clasificador un nivel de confiabilidad o rendimiento igual a 0.9867.

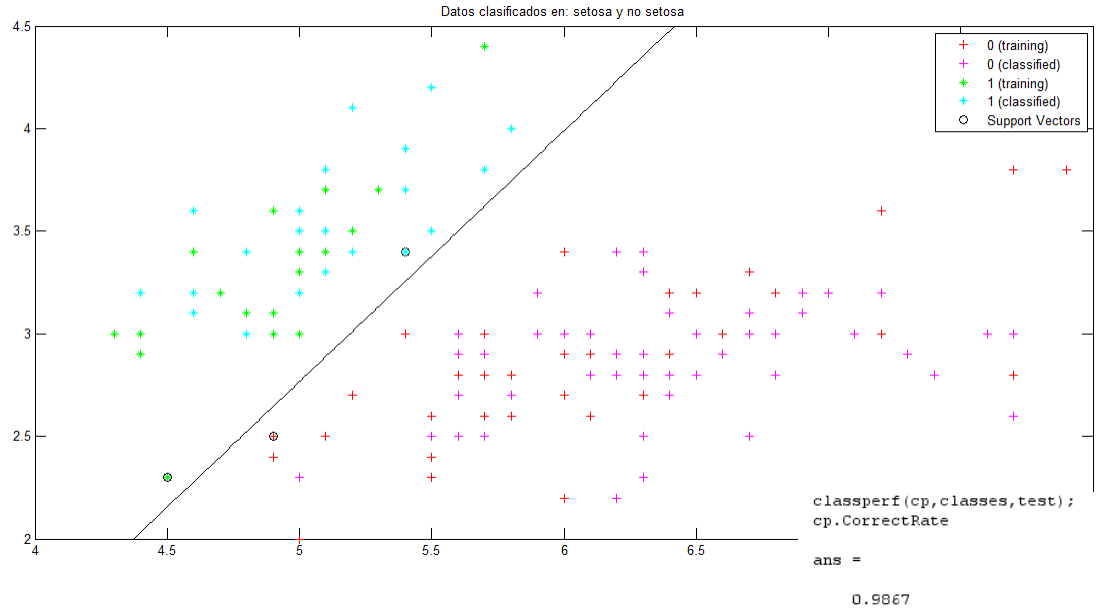


Figura 3.8 Ejemplo de clasificación por SVM's utilizando un kernel de tipo lineal [3]

3.3 Conclusiones

El uso de la clasificación jerárquica proporciona una solución al problema planteado en este trabajo escrito, puesto que permite la combinación de información entre grupos de frames que mantienen una relación jerárquica, definida por cuan tan parecidos son entre sí cada uno de los frames y seleccionando únicamente un frame dentro de cada clase obtenida (aquel que mejor representa a un grupo de frames), el cuál caracteriza a una escena principal del video en cuestión. El uso de SVM's apoya para éste trabajo de tesis, pensando en la proyección del hecho de tener que discernir entre un centro de clúster (escena principal) y una imagen almacenada en una previa base de datos, esto con la finalidad de proporcionar información ya conocida respecto a dicha imagen.

[3] Imagen capturada a partir del uso del demo Signal Processing Toolbox-Matlab- SVM's.

Capítulo 4

Aplicaciones al indexado MPEG-7

Como ya se ha planteado anteriormente, es indudable la importancia que juega el material audiovisual hoy en día, donde este tipo de material es percibido por las personas -en términos de los sentidos- con un mayor grado de completitud. La recuperación de la información sobre sistemas multimedia puede ser distinta, puesto que las personas consideran nuevos factores y asocian nuevos conceptos en sus búsquedas. De la mano de esto, la forma en que la sociedad se conecta a la red, ha cambiado radicalmente en los últimos años. Esto debido en parte, a la gran presencia que están adquiriendo los llamados teléfonos inteligentes o smartphones y tabletas, los cuales han mejorado la calidad en la conexión a internet, mientras que sus precios se han disminuido drásticamente, por lo que son de fácil adquisición, además que el hecho de que estos sean dispositivos multifuncionales, da cabida a la potencialización de una tarea, búsqueda o consulta que con ellos se lleve a cabo. Lo anterior ha apuntalado a que los dispositivos móviles se hayan convertido en uno de los principales medios de conexión a la red, siendo ya una verdadera alternativa a las formas tradicionales. Esto da origen a la aparición de las aplicaciones móviles, las cuales son aplicaciones informáticas o pequeños programas, diseñados para ejecutarse en teléfonos inteligentes, tabletas y otros dispositivos móviles, con propósitos de identificación, búsqueda o consulta, interacción, entretenimiento, entre otros. Dichas aplicaciones móviles, enriquecen al propio teléfono inteligente, puesto que crece el abanico de utilidad de éste.

La primer parte del planteamiento antepuesto, es resuelto en éste trabajo de tesis bajo la perspectiva del estándar MPEG 7 (sección 1.1), creando así un sistema que a partir de un vídeo capturado mediante un dispositivo móvil sea capaz de identificar las escenas principales de dicho vídeo. En el presente capítulo, se ejemplifican algunas aplicaciones al indexado al estándar MPEG 7, las cuales básicamente permiten la identificación de imágenes, indexación de video y reconocimiento de audio. No obstante, teniendo en cuenta la relevancia

que las aplicaciones móviles tienen hoy en día, éste trabajo se ve motivado a proponer una aplicación móvil para el Sistema Operativo Android, el cual cuenta con una gran aceptación y fuerza en el tan reciente mercado. La creación de dicha aplicación tiene como objetivo el pronto reconocimiento de entornos a través de un video capturado desde un teléfono celular, ya que al hacer uso de esta aplicación celular, la tarea resulta mucho más específica y entendible para el usuario que desea obtener información de su entorno en una determinada posición geográfica, pensada ésta última en lugares turísticos o muy concurridos.

4.1 Un descriptor para la recuperación de imágenes a gran escala basado en el bosquejo líneas características (*A descriptor for large scale image retrieval based on sketched feature lines*)

Dicha aplicación afronta el problema en el que a partir de una gran cantidad de bosquejos basados en la recuperación de imágenes, buscando en una base de datos de más de un millón de imágenes. La búsqueda está basada en un descriptor que relacione la simetría de un bosquejo binario de una imagen y una imagen completamente a color, de tal forma que estos coincidan exactamente. También se cuenta con una versión adaptada a descriptores MPEG 7 en donde el bosquejo de imagen es comparada en una base de datos de más de 1.5 millones de imágenes. Las mejores comparaciones entre el bosquejo y las imágenes de la base de datos, se consiguen por medio del histograma a color, consiguiendo compensar la ausencia de color en la consulta (bosquejo). El sistema permite a los usuarios un acceso intuitivo a grandes bases de datos de imágenes.

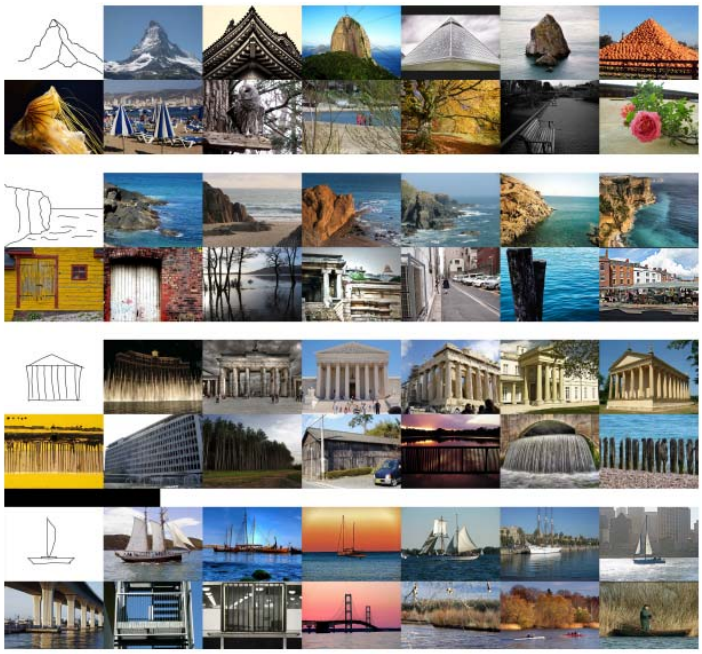
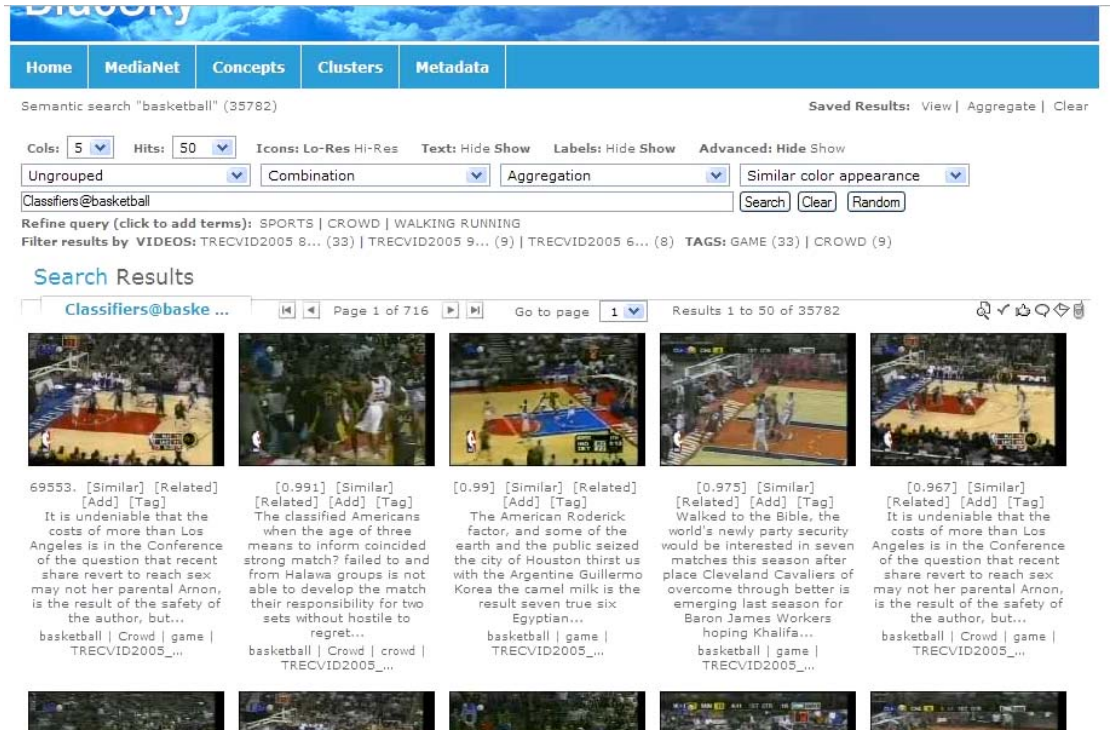


Figura 4.1 Muestra los resultados de la consulta del bosquejo de imagen (parte superior izquierda de una fila) dentro de una base de datos de 50 imágenes. En la fila superior se muestran los resultados "esperados", mientras que los "inesperados" corresponden a la fila inferior. [4]

[4] Imagen tomada del paper "*A descriptor for large scale image retrieval based on sketched feature lines*" [41]

4.2 IBM watson

Es un buscador propiamente de IBM, el cual desde hace algún tiempo se basa en lineamientos MPEG 7. Básicamente permite la consulta, clasificación y taxonomía de vídeo dentro una gran base de videos de noticias, esto a través de una imagen o precisando la búsqueda mediante texto. De igual forma se pueden ejecutar búsquedas específicas dentro de un video, tal como lo son objetos o características definidas a localizar dentro del mismo. Los resultados de los videos obtenidos a partir de los datos de la consulta, muestran información de dicho video resultante, tal como fecha de captura, título del video, entre otros datos.



4.2 Videos resultantes de la clasificación de objetos de basketball [5]

[5] Imagen tomada de la página IBM Multimedia Analysis and Retrieval demo page [42]

4.3 Shazam

Shazam es una aplicación para telefonía móvil que incorpora un servicio que permite la identificación de música. Ésta aplicación hace uso del micrófono que llevan incorporados la mayoría de teléfonos móviles para poder grabar una muestra de música que se esté reproduciendo. A partir de la muestra, se crea una huella digital acústica[6] –basadas en el espectrograma- y se compara con una base de datos de más de 11 millones de canciones[7], para encontrar coincidencias. Una vez hecha la relación, el usuario puede recibir información tal como el título de la canción, artista, álbum, enlaces de interés a servicios como iTunes, Youtube, Spotifyo Zune, así como también sugerencias de otras canciones relacionadas.

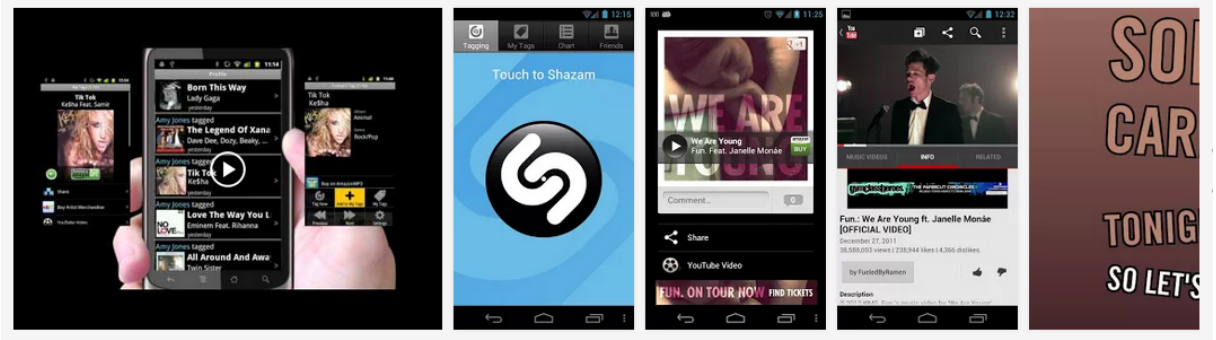


Figura 4.3 Aplicación Shazam para Android[8]

[6] Referencia *"How does shazam recognize song"* [42]

[7] Dato consultado en http://www.shazam.com/music/web/about.html

[8] Imagen obtenida de googlePlay https://play.google.com/store/apps/details?id=com.shazam.android&hl=es_419

4.4 Propuesta de aplicación móvil para la identificación de imágenes haciendo uso de datos GPS

Los dispositivos móviles inteligentes hoy en día se basan en sistemas operativos, los cuales administran todas y cada una de las aplicaciones, y en la mayoría de los casos estos cuentan con un lenguaje de programación para realizar sus aplicaciones. Los más populares sistemas operativos para la gran variedad de aparatos que existen son Android y el iOS de Mac, por lo que existen mayor número de herramientas, ayuda y conocimiento compartido en la red, por lo que dicha propuesta de aplicación móvil se enfoca en el sistema operativo Android.

La programación y funcionalidad de dicha aplicación móvil está determinada por aquellas tareas que deseamos realice. Teniendo en cuenta lo anterior y anteponiendo la parte medular de éste trabajo de tesis, es decir el sistema computacional conseguido para la identificación de escenas a partir de un video, se propone una aplicación móvil de identificación de escenas que permita integrar el sistema computacional y la practicidad del mismo, para lo cual una aplicación programada en un dispositivo celular permitirá capturar un video de mediana calidad de objetos, esta información más los datos GPS (Global Position System) del usuario serán enviados a un servidor para fines de identificación.

Para esto, se hace uso únicamente el uso de la cámara de video del dispositivo móvil y la obtención de datos GPS -esto con la finalidad de limitar la búsqueda de imágenes en la base de datos alojada en un servidor- por lo que únicamente resulta importante la ejecución de dichas herramientas, así como tener conocimiento respecto a los formatos que Android maneja para la mismas.

Una búsqueda visual móvil, se encuentra definida principalmente en tres categorías: a)búsqueda del mismo objeto tal como la imagen de la consulta proporcionada por el usuario, b) la búsqueda mediante el mismo concepto que el de la imagen en consulta, c) búsqueda por códigos o caracteres contenidos en la imagen en consulta. El principal problema del cómputo visual se centra en los descriptores de imagen, pues estos deben ser capaces de extraer estrechas características de la imagen para reconocer a la misma. Para ello, varios de los sistemas ya mencionados, utilizan el siguiente diseño típico para un sistema multimodal de búsqueda móvil presentado en la Figuras 4.4 y 4.5.

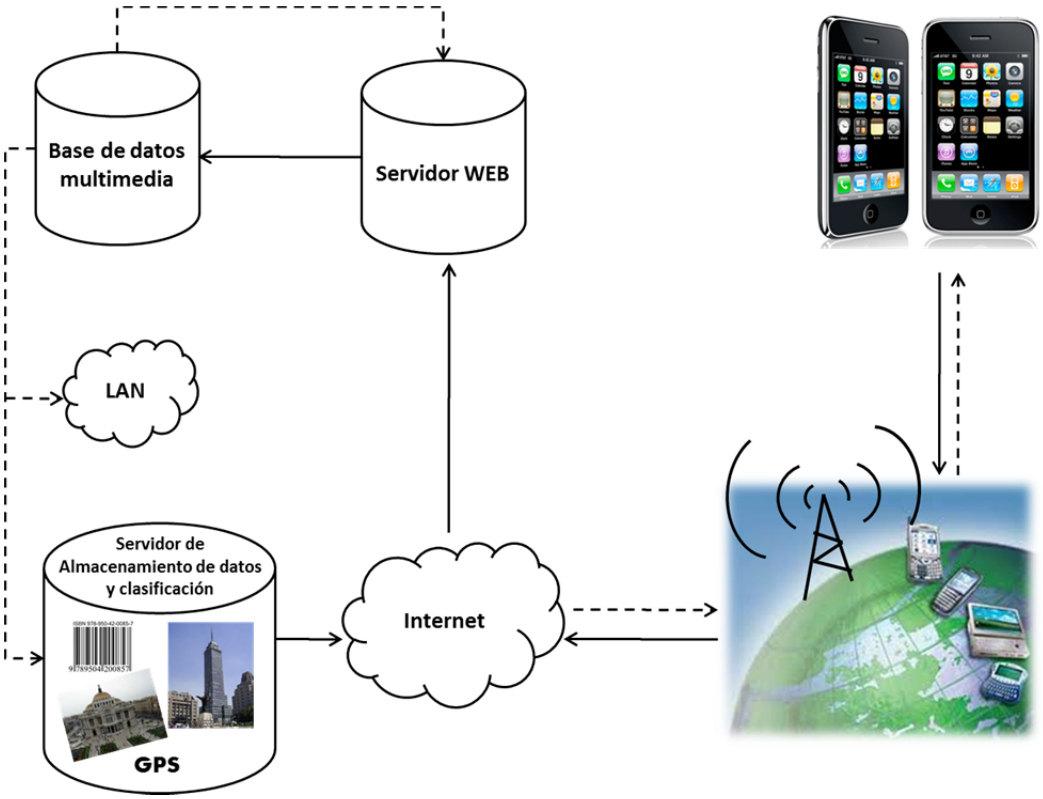


Figura 4.4. Modelo gráfico de un sistema multimodal de búsqueda móvil

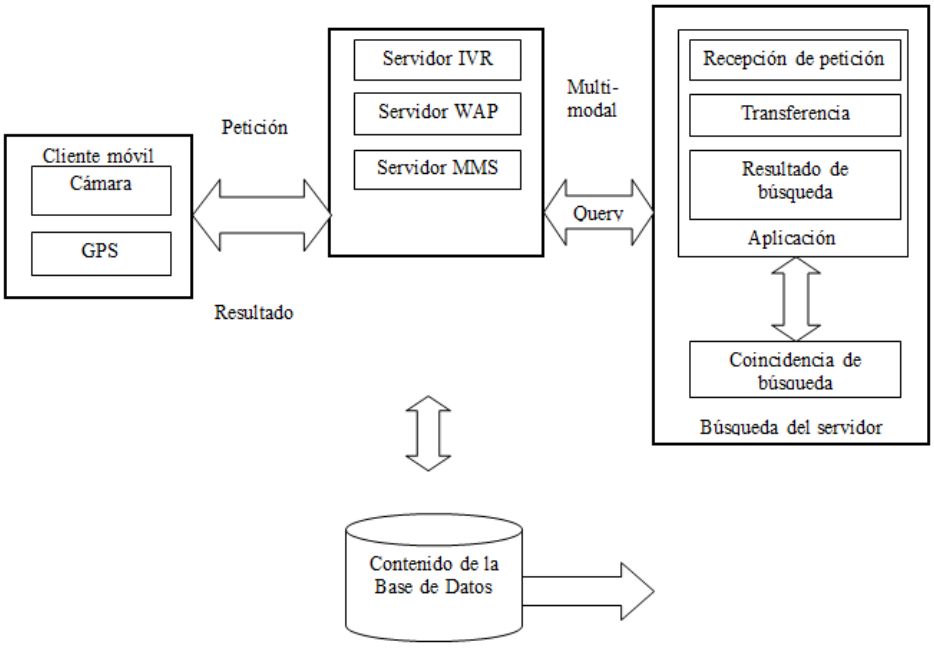


Figura 4.5. Modelo detallado de un sistema multimodal de búsqueda móvil

El esquema para ésta aplicación se compone principalmente por una captura de video, ésta se realiza al usuario grabar la secuencia de video que desea analizar para obtener las escenas más representativas del mismo. Ésta tarea aparentemente sencilla, conlleva una gran capacidad de cómputo, por lo que la aplicación móvil aquí planteada, únicamente se encargará de capturar el video y enviarla a un servidor, más los datos GPS, puesto que el dispositivo no cuenta con gran capacidad de cálculo, haciendo que su cómputo sea lento o simplemente no funcione, por ende la aplicación solo se dedicara a grabar el video, y desplegar información de los resultados en la pantalla, haciendo de esta manera una aplicación muy sencilla y al alcance de cualquier dispositivo (figura 4.6)

Figura 4.6 Esquema de aplicación para la captura de video en sistema Android 2.3.

4.4.1 Descripción de captura de video para Android

La ejecución de nuestro sistema requiere de archivos de video originales, en otras palabras el archivo .mp4 que es generado comúnmente por cualquier computadora o algunos dispositivos. Sin embargo muchos de los teléfonos no tienen la opción de generar este tipo de

archivos de video, en cambio son codificados en otros tipos como son el conocido .avi. .mpeg, .mov, entre otros. No obstante, esto no es un grave problema para el sistema, ya que éste es capaz de realizar las conversiones pertinentes para que dicho video se analizado en formato .mp4.

En la figura 4.7 podemos observar un fragmento de código que permite realizar la captura de un video para Android proporcionando la información de codificación para el archivo.

```
package com.example;

import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;

import android.content.Context;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.hardware.Camera;
import android.hardware.Camera.PreviewCallback;
import android.util.Log;
import android.view.SurfaceHolder;
import android.view.SurfaceView;

class Preview extends SurfaceView implements
SurfaceHolder.Callback {
	private static final String TAG = "Preview";

	SurfaceHolder mHolder;
	public Camera camera;

	Preview(Context context) {
		super(context);

		// Install a SurfaceHolder.Callback so we get
notified when the
		// underlying surface is created and destroyed.
		mHolder = getHolder();
		mHolder.addCallback(this);

	mHolder.setType(SurfaceHolder.SURFACE_TYPE_PUSH_BUFFERS);
	}
```

Figura 4.7 Código en Android para captura de video

En el código se especifica o se crea un objeto llamado Camera este objeto hace referencia al hardware (cámara) que llevara la acción de grabación de video. En primera instancia se accede a la cámara del dispositivo, en dónde ello conlleva a la apertura y cierre del obturador (inicio y fin de captura) y al micrófono del dispositivo (puesto que se trata de video), para posteriormente dar paso a los formatos de salida y un codificador de video.

El ejemplo el formato especificado para la aplicación es .mp4 que es un formato muy común para la captura de video en teléfonos celulares smartphones, no obstante el sistema cuenta como antes mencionado diversos formatos que pueden ser utilizados para la codificación del video.

Formatos de Video Android (Extensión del Archivo)	Códec	Nota
3GPP (.3gp) MPEG-4 (.mp4)	H.263	
3GPP (.3 gp) MPEG-4 (. mp4) MPEG-TS (. ts)	H.264 AVC	sólo MPEG-TS con audio AAClo
3GPP (.3 gp)	MPEG-4 SP	
WebM (. WebM) Matroska (. mkv, Android 4.0 +)	VP8	Reproducible sólo en Android 4.0 y superior.

Figura 4.8 Formatos de codificación de video soportados por Android

En la figura 4.8 tenemos una lista de formatos de salida para codificación de video utilizada para el sistema operativo Android, de igual manera para el sistema es manejado de manera similar para obtener el formato deseado.

Los sistemas operativos Android e iOS ambos ocupan la codificación MP4 aunque pueden ser configurados en ambos para obtener los parámetros deseados para su la captura de video.

4.4.2 Envío de datos de video y de respuesta.

Los datos capturados por el usuario (video y datos GPS) y los datos de respuesta (escenas) están pensados para ser enviados vía internet o en su defecto un mensaje multimedia siendo esta una opción para aquellos que no cuenten con acceso a la red. Esto puede dificultar la tarea dependiendo de las características que el video posee, por lo que éstas se ven limitadas a la captura de videos cortos (10 a 18 segundos) aún en HD (High Definition).

Una de las características que debemos tomar en cuenta es que la aplicación debe tener el permiso para tener la capacidad de guardar el archivo de video (memoria interna o externa del dispositivo) para posteriormente ser enviadas, claro está que también debe ser especificado el puerto por el cual el archivo se va enviar y a recibir.

Los datos serán primeramente recibidos por un servidor el cual procesará inmediatamente el video obteniendo las escenas principales del mismo, capturando cada una de estas escenas en una imagen representativa, mismas que serán enviadas al usuario.

4.5 Conclusiones

Hoy en día MPEG 7 juega un papel revolucionario en cuanto a la forma en que se guardan los datos, puesto que estos ya no son sólo bits, sino representan características o la descripción de los mismos. Valiéndose de MPEG 7 y de aplicaciones móviles, se logra potencializar diversas tareas de la vida cotidiana, además de que para el usuario, éstas se vuelven mucho más sencillas de ejecutar. Es importante tomar en consideración la usabilidad y el fin de cada aplicación móvil, para de ésta forma conocer los hábitos y necesidades de cada usuario. Las aplicaciones móviles no son estándares, éstas dependen del sistema operativo bajo la cual se pongan a prueba. Dentro de las principales ventajas que conlleva el uso de aplicaciones móviles, es el poder explotar al máximo las capacidades de los mismos, como lo son las funcionalidades GPS, localización, bluetooth, cámara, micrófono, marcación rápida, etc.. Dichas herramientas podrían potencializar la implementación de una tarea, puesto que proporcionan datos y ayuda extra; además de que para el usuario resulta mucho más rápido, amigable y portable el hecho de que con un sólo dispositivo, pueda efectuar quehaceres de su día a día.

Pensar en la implementación de una aplicación móvil que en la actualidad no se encuentra en el mercado y podría resultar muy útil para la identificación de escenas, objetos en movimiento y en general la indexación de video, resulta ser una proyección muy ambiciosa. Para la implementación de ésta se necesita la optimización de rápidos y robustos algoritmos, sin embargo éste es un campo prometedor de estudio.

Capítulo 5

Evaluación y resultados del agrupamiento

El objetivo de los métodos desarrollados en los capítulos anteriores es proporcionar un buen descriptor escalable y un método de clasificación de las principales escenas de un video en general. El objetivo de éste capítulo es someter a prueba el descriptor y el método de clasificación, para así conocer las capacidades que estos tienen ante los diversos escenarios contenidos en un video. El primer paso a realizar, es la medición de la similitud entre frames para con ello realizar la clasificación de escenas por medio de un clustering jerárquico.

5.1 Características de video

La base de datos de video ha sido capturada mediante dos tipos de dispositivos: un teléfono celular con cámara de video y una cámara digital convencional. Los videos capturados mediante el celular son todos ellos en formato mp4 con diversas características en tamaño y tasa de fotogramas por segundo (*FPS- Frames Per Second*). Los videos tomados desde la cámara digital se encuentran en formato mp4 con un FPS igual a 14 y una dimensión de 720x540 píxeles por fotograma.

5.2 Procedimiento de descripción de la imagen

El descriptor propuesto tiene la característica de ser multiresolución, lo que nos permite escoger el nivel de resolución dentro de la pirámide de ondas que va a ser utilizado para la comparación. Ésta investigación y trabajo experimentado, contempla únicamente el mismo nivel de comparación entre las imágenes, es decir, la $imagen_i$ y la $imagen_j$ corresponden al mismo nivel k de descomposición Wavelet. De igual forma, para efectuar la comparación entre frames, son contemplados todos los coeficientes obtenidos a partir de la Transformada Discreta de Wavelet. Para cada imagen Wavelet se efectúa un histograma en bajo nivel (ec. 2.2) y un histograma en alto nivel (ec. 2.3), con el fin de realizar pruebas a partir de una imagen compresa (subbanda LL) y pruebas a partir de datos de textura (subbanda HF).

Para el clustering jerárquico aglomerativo, para cada video son realizadas pruebas con 4 diferentes valores "n" de escenas calculadas: 3, 5, 7 y 10 (centros de cluster), es decir, un $n=3$ define la obtención de únicamente de 3 escenas principales del video, con un n=5 se consiguen 5 escenas principales del video, etc.

5.3 Evaluación de la clasificación

La evaluación está dada a partir de la visualización manual de un video e identificación de las principales escenas según el observador. La figura 5.1 describe de manera gráfica el posicionamiento de una persona en la zona centro del Distrito Federal, en dónde éste se dispone hacer una captura de video en forma de barrido (vista panorámica). Como se puede apreciar, visualmente se identifican por lo menos 4 escenarios principales: el monumento Hemiciclo a Juárez, la Alameda Central, el Palacio de Bellas Artes y la Torre Latinoamericana.

Figura 5.1 Descripción gráfica de la ubicación de un usuario ante la grabación de un video[9]

Así bien, una vez capturado el video, la tarea consiste en poder identificar esas escenas visuales a través de sistema aquí propuesto. En la figura 5.2 se describe por medio de un diagrama la metodología a seguir para cumplir con dicha tarea.

[9] Mapa tomado de la página de internet turística http://www.mexicofirstclass.com

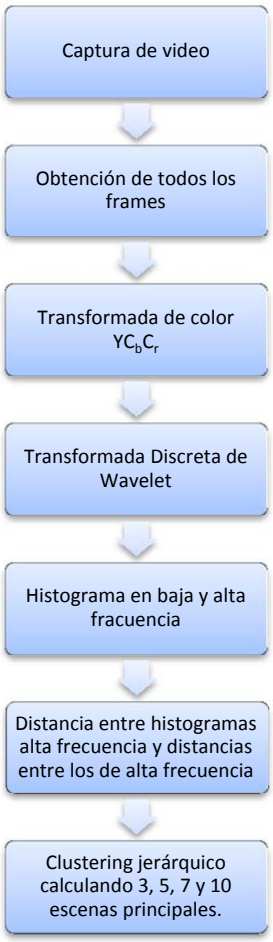


Figura 5.3 Metodología para obtener las escenas principales de un video

Video “Edificios y área verde”

Duración: 15 segundos
Formato: mp4
Tasa: 14 frames por segundo
Dimensión: 320 x 240 píxeles

Una vez observado el video, las principales escenas principales que se identifican, son:

Al efectuar la metodología propuesta (figura 5.3), los resultados obtenidos se muestran en las siguientes tablas:

Calculando 3 escenas principales (con coeficientes LF)			
Número de escena	**Número de frames agrupados**	**Centro de Cluster**	**Imagen (centro de cluster)**
1	95	Imagen47	
2	23	Imagen107	
3	116	Imagen178	

TAB 5 Resultados del video “Edificios y área verde” para 3 centros de cluster, tomando en cuenta coeficientes de baja frecuencia.

Calculando 5 escenas principales (con coeficientes LF)			
Número de escena	**Número de frames agrupados**	**Centro de Cluster**	**Imagen (centro de cluster)**
1	69	Imagen35	
2	57	Imagen97	
3	11	Imagen131	
4	36	Imagen155	
5	62	Imagen204	

TAB 5.1 Resultados del video “Edificios y área verde” para 5 centros de cluster, tomando en cuenta coeficientes de baja frecuencia.

Calculando 7 escenas principales (con coeficientes LF)			
Número de escena	**Número de frames agrupados**	**Centro de Cluster**	**Imagen (centro de cluster)**
1	11	Imagen6	
2	28	Imagen26	
3	50	Imagen65	
4	3	Imagen91	
5	33	Imagen109	

6	57	Imagen154	
7	51	Imagen209	

TAB 5.2 Resultados del video “Edificios y área verde” para 7 centros de cluster, tomando en cuenta coeficientes de baja frecuencia.

Calculando 10 escenas principales (con coeficientes LF)			
Número de escena	**Número de frames agrupados**	**Centro de Cluster**	**Imagen (centro de cluster)**
1	35	Imagen18	
2	32	Imagen52	
3	3	Imagen69	

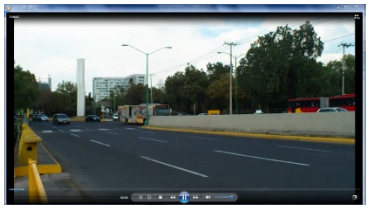
4	22	Imagen82	
5	36	Imagen111	
6	33	Imagen145	
7	8	Imagen166	
8	23	Imagen181	
9	29	Imagen207	

10	13	Imagen228	

TAB 5.4 Resultados del video “Edificios y área verde” para 10 centros de cluster, tomando en cuenta coeficientes de baja frecuencia.

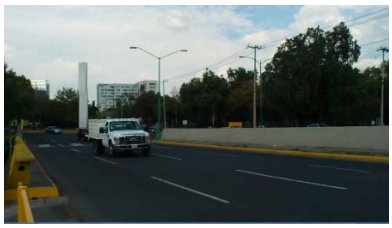
Video “Avenida Insurgentes”

Duración: 16 segundos
Formato: mp4
Tasa: 29 frames por segundo
Dimensión: 1280 x 720 píxeles



Una vez observado el video, las principales escenas principales que se identifican son:

Al efectuar la metodología propuesta (figura 5. 3), los resultados obtenidos se muestran en las siguientes tablas:

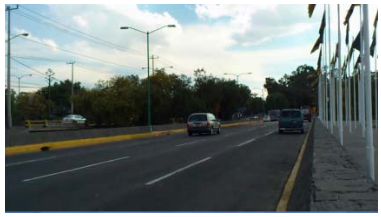
Para 3 escenas principales (con coeficientes LF)			
Número de escena	**Número de frames agrupados**	**Centro de Cluster**	**Imagen (centro de cluster)**
1	6	Imagen4	
2	402	Imagen208	
3	65	Imagen441	

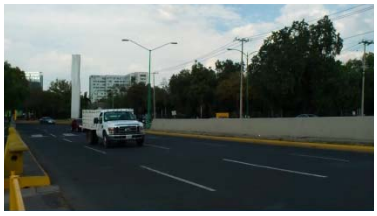
TAB 6. Resultados del video “Avenida Insurgentes” para 3 centros de cluster, tomando en cuenta coeficientes de baja frecuencia.

Para 5 escenas principales (con coeficientes LF)			
Número de escena	**Número de frames agrupados**	**Centro de Cluster**	**Imagen (centro de cluster)**

1	27	Imagen14	
2	51	Imagen53	
3	255	Imagen206	
4	134	Imagen401	
5	6	Imagen471	

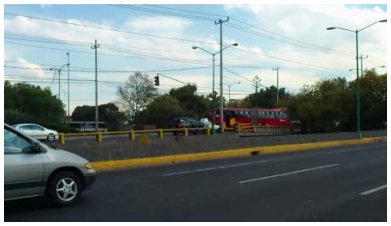
TAB 6.1 Resultados del video “Avenida Insurgentes” para 5 centros de cluster, tomando en cuenta coeficientes de baja frecuencia.

Para 7 escenas principales (con coeficientes LF)			
Número de escena	**Número de frames agrupados**	**Centro de Cluster**	**Imagen (centro de cluster)**
1	129	Imagen65	
2	47	Imagen153	
3	25	Imagen189	
4	96	Imagen250	
5	9	Imagen302	

6	164	Imagen389	
7	3	Imagen372	

TAB 6.2 Resultados del video “Avenida Insurgentes” para 7 centros de cluster, tomando en cuenta coeficientes de baja frecuencia.

Para 10 escenas principales (con coeficientes LF)			
Número escena	**Número de frames agrupados**	**Centro de Cluster**	**Imagen (centro de cluster)**
1	8	Imagen5	
2	72	Imagen45	
3	46	104	

4	3	Imagen128	
5	106	Imagen183	
6	81	Imagen276	
7	84	Imagen359	
8	16	Imagen409	
9	36	Imagen435	

10	21	Imagen463	

TAB 6.3 Resultados del video "Avenida Insurgentes" para 10 centros de cluster, tomando en cuenta coeficientes de baja frecuencia.

Observando las tablas en el intervalo de la TAB. 5 a la TAB. 6, podemos notar que la obtención de las escenas principales de los videos se encuentran definidas experimentalmente por el cluster jerárquico aglomerativo efectuado con 5 centros de cluster, arrojando resultados favorables, pues dichas escenas encontradas experimentalmente bajo el método descrito previamente en las secciones anteriores de éste trabajo, corresponden en buena medida a las escenas que visualmente se han elegido como las más relevantes en cada uno de los videos. Las imágenes conseguidas con centros de cluster mayores a 7, resultan un tanto redundantes en cuanto a escenas y en el caso contrario, es decir, con centros de cluster menores a 5, las imágenes aquí obtenidas resultan un tanto pobres en cuanto a la elección de las escenas principales del video en cuestión.

5.4 Conclusiones

La fase experimental presentada en ésta sección nos permite validar el método elegido para la obtención de las escenas principales de un video. Los resultados aquí presentados, indican que la selección de las imágenes que mejor representan una escena en una secuencia de video, es altamente buena, pues en promedio con el cálculo de 5 keyframes conseguimos puntualizar los escenarios de una secuencia de video. Los resultados han demostrado que el descriptor propuesto es eficiente y robusto a la calidad de la segmentación de video.

Capítulo 6

Conclusiones y perspectiva

En el presente trabajo expone el estudio de un método o sistema que lleva a la obtención de las escenas principales de un video, haciendo uso de descriptores visuales y de textura, compuesto por cuatro etapas: descomposición wavelet, cálculo del histograma, distancias entre histogramas y clasificación de imágenes.

A pesar de la dificultad de sugerir una elección de los descriptores aquí empleados, esto dependiendo de las tareas a realizar, éste trabajo logra conseguir reunir un descriptor local de bordes y textura mediante el uso de Wavelets y un descriptor global haciendo uso del Histograma Wavelet en baja y alta frecuencia.

La importancia de la primera etapa radica en la obtención de imágenes compactas respecto a la original, sin embargo de igual forma se consigue trabajar con información más relevante de la imagen, ya que el uso de las wavelets nos brinda información de bordes y textura en cada una de las diferentes subbandas. El hecho de diseñar un histograma en alta frecuencia nos permite tener agrupada toda la información de bordes y textura (coeficientes wavelets en las subbandas LH, HL y HH) pertenecientes a un misma imagen, para de ésta forma ser utilizado como el principal descriptor de cada frame del video. En la tercera etapa de esta investigación, se consigue proponer una medida de disimilitud entre histogramas, es decir, entre aquello que caracteriza a una imagen, dicha propuesta reside en comparar uno a uno cada coeficiente wavelet de todas las imágenes que conforman el video, esto bajo el algoritmo de Chi-square. Finalmente la clasificación de imágenes resulta dada por el clustering jerárquico aglomerativo, el cual queda definido principalmente por la medida de disimilitud Chi-square y el enlace entre clusters del tipo mean-link. En ésta última etapa los resultados obtenidos son muy favorables; generalmente los videos utilizados para las pruebas experimentales cuentan

con una duración entre 15 y 17 segundos, con diversas características de formato, definición, dimensión y tasa. Los resultados muestran que en promedio las escenas principales de cada video, quedan definidas por 5 centros de cluster. Dichos centros de cluster al ser comparados con las escenas que visualmente podemos definir, resultan ser las mismas en un porcentaje mayor al 90%. Aún en videos de larga duración (mayores a un minuto), se consiguen definir las principales escenas en no más de 10 frames, escenas que comparadas con las clasificadas visualmente, prácticamente resultan ser las mismas. Cumpliendo de ésta forma satisfactoriamente con el objetivo de la tesis aquí propuesta.

El costo computacional que requieren las cuatro etapas anteriormente descritas, es alto. La primera etapa computacional (descomposición Wavelet) requiere de una gran capacidad de almacenamiento pues por cada frame se obtienen cuatro imágenes más, lo cual se traduce en casi la mitad del espacio de almacenamiento requerido por la extracción total de los frames de cada video. El tiempo de procesamiento en ésta etapa es de aproximadamente 5 segundos por frame. Para la segunda etapa, el cálculo del histograma no resulta tan costoso, el espacio de almacenamiento es mínimo y el tiempo de ejecución es aproximadamente igual a 3 segundos por histograma calculado paralelamente (el histograma en baja y alta frecuencia son calculados paralelamente). La tercera etapa resulta ser la pequeña en almacenamiento, sin embargo es la más tardada en ejecución, puesto que se requiere aproximadamente de 0.03 segundos por comparación de histogramas. Para la etapa final correspondiente al clustering jerárquico, el tiempo de ejecución resulta ser de unos cuantos segundos dependiendo del tamaño de la matriz de distancias. Lo anterior define el problema fundamental de ejecutar dicho sistema por medio de una aplicación celular, en dónde los resultados en tiempo real deben ser obligatoriamente casi automáticos, sin embargo el costo computacional que refleja nuestra experimentación, la idea de una implementación para dispositivos móviles no resulta viable, dejando así abierta dicha posibilidad a trabajos futuros.

Éste trabajo se ve limitado al reconocimiento de objetos en movimiento, puesto que lo único que se reconoce en las escenas de un video es el fondo o el escenario, por lo que ésta resulta ser una proyección del mismo. Ésta línea de investigación propone aproximaciones para capturar las características importantes espacio-temporal de las acciones, lo que generalmente requiere una fase de aprendizaje muy amplia. Poniendo un ejemplo sencillo, podemos imaginar un video capturado en una playa, en donde algunos toman el sol, otros practican el surf y unos más caminan en la playa, ¿podría automáticamente una computadora

decirnos lo que pasa en cada una de las escenas?, ¿sería capaz de identificar las diferentes acciones humanas?. Actualmente existe un robusto campo de investigación en la detección y la clasificación de la acción humana en secuencias de video. Dicha tarea de la clasificación automática y la localización de acciones humanas o seguimiento de objetos en secuencias de video es sumamente interesante para una gran variedad de aplicaciones, como en video vigilancia, estudio de las acciones de pacientes con enfermedades de demencia [34], entre otros. Por lo general en éste tema de investigación, existen tres tipos de rasgos que caracterizan o describen la postura y el movimiento: rasgos estáticos basados en bordes y forma [36], rasgos dinámicos basados en flujos ópticos [37] y los rasgos espacio-temporales [38]. Éstas últimas han demostrado ser especialmente prometedoras en la marcha para la compresión de movimiento debido a su rico poder descriptivo. En la publicación "A Hierarchical Model of Shape and Appearance for human Action Classification" [39] se presenta un modelo para caracterizar las acciones humanas, en particular se propone un modelo que tiene en cuenta tanto rasgos estáticos como dinámicos del movimiento humano. De igual forma existen publicaciones que proponen algoritmos capaces de decidir qué acción humana se encuentra presente en una secuencia de video de acuerdo a aprendizaje adquirido [40].

Así bien, gracias al trabajo de estancia realizado en el laboratorio de investigación informática de Bordeaux (LaBRI), se consiguió refinar el método a emplear para conseguir el objetivo de éste trabajo de investigación, con la colaboración y supervisión de la Professeur Jenny Benois-Pineau, en donde se nos facilitó material escrito de previas investigaciones, el préstamo de equipo de cómputo adecuado para efectuar las pruebas necesarias, así como también código bajo el lenguaje C++, el cual conjuntamente con nuestro propio código elaborado, fue utilizado para los experimentos efectuados en el mismo laboratorio.

El área de investigación resulta ser muy prometedora, sí bien lo aquí desarrollado ha sido una pequeña fracción dentro del ámbito de segmentación de video, la tarea de reconocimiento de acciones dentro de un video resulta ser bastante difícil. Actualmente es un aspecto pendiente y en desarrollo en el rubro del procesamiento de imágenes y video. Mejores métodos han de ser desarrollados todavía, para conseguir resultados mayormente útiles, eficientes y a menor costo computacional dentro de éste campo.

Referencias

[1] Y. Watanabe, K. Sono, K. Yokoizo, and Y. Okada, BTranslation camera on mobile phone, in Proc. IEEE Int. Conf. Multimedia Expo, Baltimore, MD, julio 2003.

[2] M. Smith, D. Duncan, and H. Howard, AURA: A mobile platform for object and location annotation, in Proc. 5th Int.Conf. Ubiquitous Computing, Seattle, WA, octubre 2003.

[3] M. Flickner, H. Sawhney, W. Niblack et al., *"Query by image and video content: The QBIC system ",* IEEE Computer, Special Issue on Content-Based Retrieval, septiembre 1995.

[4] J. R. Smith and S.-F. Chang, *"VisualSEEk: A fully automated content-based image query system",* in Proc 4th ACM Int. Conf. Multimedia, Boston, MA, noviembre 1996.

[5] N. Dimitrova, H.-J. Zhang, B. Shahraray, I. Sezan, et A. Huang, T.andZakhor., "*Applications of video content analysis and retrieval*", IEEE Mutimedia, julio 2002.

[6] (JPEG2000) ISO/IEC 15444-1 :2004, Information technology. *"JPEG2000 image coding system* ",part 1.

[7] G. Sullivan. *"Efficient scalar quantization of exponential and laplacian variables"*. IEEE Trans. diciembre 1996.

[8] A. Said and W.A. Pearlman. *"A new fast and efficient image codec based on set partitioning in hierarchical trees"*. IEEE Trans. Circuits Syst. Video Technol, junio 1996.

[9] J.M. Shapiro. *"Embedded image coding using zero trees of wavelet coefficients"*. IEEE Trans. Signal Processing, 1993.

[10] D. Taubman. *"High perfomance scalable image compression with ebcot"*. IEEE Trans. Image Processing, julio 2000.

[11] M. C. Angelides, H. Agius, J. Zhang. L. Ye, J. Ma,"The Handbook of MPEG Application: Standards in Practice", Edit. WILEY, 2011.

[12] J. Ruíz, *"Introducción al estándar MPEG-7"*, Universidad Politécnica de Cataluña, Image processing Group, TSC Departament, 2006.

[13] P.J. Vivancos, *"El estándar MPEG-7"***,** InforMAS revista de Ingenieria Informática del CIIRM, Ilustre Colegio de ingenieros en informática de la Región de Murcia, Murcia , julio 2005.

[14] S. Mallat., *"A Wavelet Tour of Signal Processing"*, Academic Press, 1998.

[15] C.R. Jung., *"Multiscale image segmentation using wavelets and watersheds",*IEEE Proceedings of the XVI Brazilian symposium on computer graphics and image processing, p.p. 278–283, 2003.

[16] J.B. Kim, H.J. Kim., *"Multiresolution-based watersheds for efficient image segmentation. Pattern recognition letters",* Elsevier, p.p. 473–488, 2003.

[17] (MJPEG2000) ISO/IEC 15444-3:2007.,*"Information technology jpeg2000 image coding system: Motion jpeg2000"*.

[18] J. Hafner, H.S. Sawhney, W. Equitz, M. Flickner et W. Niblack, *"Efficient color histogram ndexing for quadratic form distance function"*, IEEE Transaction on Pattern Analysis and Machine Intelligence,p.p. 381–392, 1995.

[19] S. Mallat, *"A Wavelet Tour of Signal Processing",* Academic Press*,* 1998.

[20] S. C. Johnson, *"Hierarchical Clustering Schemes",* 1967, p.p. 241-254

[21] Everitt, Landau and Leese, *"Cluster Analysis"*, 2001, sección 4.2.

[22] J. Benois-Pineau, F. Precioso, M. Cord, *"Visual Indexing and Retrieval",* France 2012.

[23] U. Gargi, S. Oswald, D. Kosiba, S. Devadiga et R. Kasturi., *"Evaluation of video sequence indexing and hierarchical video indexing"*, Proc. of SPIE Storage and retrieval in image and video databases III,p.p. 144–151.

[24] R. Kasturi, S.H. Strayer, U. Gargi, S. Antani, *"An Evaluation of Color Histogram Based Methods in Video Indexing"*, 1996.

[25] T.F. Smith, M.S. Waterman., *"Identification of common molecular subsequences",* Journal of Molecular Biology, 1981.

[26] A. Vailaya, A. Jain, H. Zhang, *"Video Clustering",* Michigan State University, 1999.

[27] H.J. Zhang and D. Zhong., *"A scheme for visual feature based image indexing"*, In Proc. SPIE Conference on Storage and Retrieval for Images and Video Databases, San Jose, CA, febrero 1995.

[28] D. Zhong, H.J. Zhang, and S.-F. Chang, *"Clustering methods for video browsing and annotation"*, In Proc. SPIE Conference on Storage and Retrieval Images for Images and Video Databases, San Jose, CA, febrero 1995.

[29] V.N. Vapnik. Statistical Learning Theory. John Wiley & Sons, Inc, 1998.

[30] S. German and E. Bienenstock. Neural networks and the bias / variance dilemma. Neural Computation, p.p. 1-58, 1992.

[31] I. Guyon, V. Vapnik, B. Boser, L. Bottou, S. Solla*. "Structural risk minimization for character recognition. Advances in Neural Information Processing System"*, p.p. 471-479, 1992.

[32] D. Montgomery, E. Peck, *"Introduction to Linear Regression Analysis"*. Edit. WILEY, 2da edición, 1992.

[33] L. Gonzáles, *"Análisis discriminante utilizando máquinas núcleo de vectores soporte. Función núcleo similitud".* Tesis doctoral, Dpto. Economíaa Aplicada I. Universidad de Sevilla, junio 2002.

[34] S. Karaman, J. Benois-Pineau, V. Dovgalecs, R. Mégret, J. Pinquier, R. André-Obrecht, Y. Gaëstel, J.-F. Dartigues, *"Hierarchical Hidden Markov Model In Detecting Activities Of Daily Living In Wearable Videos For Studies Of Dementia"*, Multimedia Tools And Applications (Article In Press), 2012.

[35] C. Morand*, "Segmentation spatio-temporelle et indexation vidéodans le domaine des représentationshiérarchiques"*, Thesis Université Bordeaux 1, France 2009.

[36] N. Dalal, B. Triggs, and C. Schmid, *"Human detection usingoriented histograms of flow and appearance"*, ECCV, 2006.

[37] A. A. Efros, A. C. Berg, G. Mori, and J. Malik., *"Recognizing action at a distance"*, In ICCV, 2003.

[38] C. Fanti, L. Zelnik-Manor, and P. Perona, *"Hybrid models for human motion recognition"*, In CVPR, 2005.

[39] J. C. Niebles, F.F. Li, *"A Hierarchica, Model of Shape and Appearance dor Human Action Classification"*, CVPR07(1-8), IEEE, 2007.

[40] J.C. Niebles, Hongcheng Wang and Li Fei-Fei, *"Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words"*, International Journal of Computer Vision (IJCV), sep. 2008.

[41]E. Mathias., K. Hilderbrand., T. Boubekeur, A. Marc, *"A descriptor for large scale image retrieval based on sketched feature lines",* EUROGRAPHICS Symposium on Sketch-Based Interfaces and Modeling, 2009.

[42] Sitio web official Shazam [online]:
http://www.soyoucode.com/2011/how-does-shazam-recognize-song

[43] Blog de programación móvil, Disponible [online]:
http://www.culturelabel.com/blog/2012/02/24/mighty-mobile/

[44] Blog de programación móvil, Disponible [online]: http://programaciontotal.blogspot.com

[45] Blackberry Desarrollo , Disponible [online]: http://us.blackberry.com/developers/