

Sistema de Indexación de Vídeo por Contenido y Transmisión en Tiempo Real

Antoni Bibiloni, Ricardo Galli, Esteve Lladó, Marcos Rodriguez
Depart. Matemàtiques i Informàtica
Universitat de les Illes Balears
Ctra. de Valldemossa Km 7,5
E-07071 Palma (Balears)

Tel: 971 173204
Fax: 971 173003
e-mail: dmiabc0@ps.uib.es

Abstract

The interest in indexing and querying video data is growing rapidly. Desktop video is a new and attractive medium for communication between people using computers and networks. The main focus in any video information system is to develop a database management system with a friendly content-based retrieval of the digital video information.

In this paper, we present an on-line video system for educational purposes. The interface consist of two main modules, An *annotation-video-interface* which can be used to put keywords, comments, proposal exercises, recommended lectures and so on in video segments. The annotations specify video sequences information for the automatic generation of a video library indexed by content. And a *query-video-interface* which can be used to extract the relevant segments of video that satisfy the specific query condition. We also describe the implementation of a real time video server, which make uses of a protocol designed for video conferencing to transmit the video sequences.

1. Introducción

Este artículo presenta detalles de implementación de un prototipo de indexación y recuperación de vídeo basado en contenido. El objetivo del proyecto es definir e implementar una base de datos de videos digitales que permita el acceso, navegación y visualización por contenido. Las secuencias de vídeo provienen de clases universitarias, seminarios y conferencias. Por ello, la base de conocimiento es sobre un dominio específico siendo suficiente el uso de palabras clave para la descripción semántica del contenido, en la sección 0 describimos el modelo de datos utilizado.

El sistema tiene tres módulos principales, un módulo de usuario para facilitar las anotaciones e indexación del vídeo (sección 0), un módulo que permite la navegación, búsqueda y recuperación basada en el contenido [1][8][9][10] (sección 0) y un sistema de servicio y transmisión en tiempo real de vídeo y audio. Cada secuencia de vídeo es indexada con ayuda del módulo de anotación utilizando la semántica extraída de sus imágenes y audio. A partir de la metainformación generada en la anotación, se generan los átomos que forman la unidad mínima de indexación. Estos átomos son almacenados en los servidores en formato Quicktime [16].

Para transmitir el vídeo Quicktime en tiempo real hemos desarrollado un sistema basado en una arquitectura cliente/servidor sobre TCP/IP que permite extraer las películas en formato QuickTime y enviarlas en tiempo real a los clientes utilizando el protocolo MovieTalk [15] (sección 0).

2. Modelo de Datos

La automatización de la descripción semántica utilizando técnicas del tratamiento de imágenes y visión por ordenador, aunque si existe una gran investigación sobre el tema de la segmentación del vídeo, todavía no son lo suficientemente satisfactorias [12][13][14]. Nuestro modelo es fuertemente dependiente de la naturaleza de la fuente de vídeo, y está diseñado para un entorno educativo. La metodología del sistema consiste en utilizar un modelo definido donde el profesor o conferenciante interpreta y transcribe la semántica de las secuencias vídeo sobre el modelo de datos.

La utilización del modelo consiste en que sólo los índices o combinación de ellos pueden ser utilizados para construir las consultas sobre la base de datos de videos indexados (sección 0). El modelo está estructurado como árboles de segmentos vídeo [2] (figura 1), manteniendo la segmentación de átomos, escenas y segmentos presentados en [1].

La definición de los términos son:

- Átomos: la unidad mínima de fotogramas consecutivos
- Escena: conjunto de átomos ordenados representando un contenido concreto.
- Segmento: unión de escenas bajo un criterio determinado.

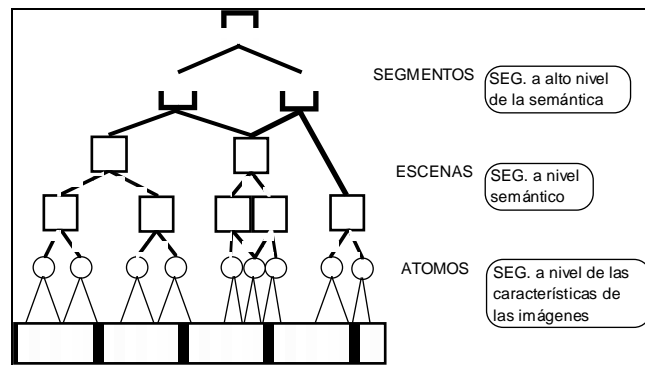


Figura 1: Árbol de segmentos.

2.1 Diseño de la Base de Datos

Este modelo de datos está implementado sobre una base de datos relacional extendida encargada de editar y actualizar las tablas de metainformación necesarios para la posterior visualización, edición y navegación del vídeo digital indexado [5]. Las tablas de metainformación almacenan los atributos de las entidades mencionadas anteriormente y las referencias a los servidores y ficheros (átomos) de vídeo en formato Quicktime.

Para describir la semántica de los vídeo educativos se han implementado cinco tablas, en ellas también se representa las relaciones, sinónimos y jerarquías de las palabras clave utilizadas para la representación del contenido.

1. (*Carrera*)
2. (*Asignatura*)
3. (*Profesor*)
4. (*Clase*)
5. (*Tema*)

CARRERA		
Codigo	N	4
Nombre	C	20
Descripción	C	Memo

PROFESOR		
Codigo_Prof	N	4
Nombre	C	40
.....		

ASIGNATURA		
Codigo_A	N	4
Codigo_C	N	4
Nombre	C	20
Curso	C	5
.....		

CLASE		
Codigo_CL	N	4
Codigo_A	N	4
Codigo_Prof	N	4
Apunt_Video	C	30
Fecha_grab	D	8
.....		

TEMA		
Codigo_T	N	4
Codigo_CL	N	4
Inicio	SMPTE	11
Fin	SMPTE	11
Descripción	C	20
Dependencia	N	4
.....		

Figure 3: Tablas Índices

3. Arquitectura general de sistema

3.1 SGBD

El SGBD almacena la metainformación referente a cada átomo de vídeo almacenado en el sistema distribuido de servidores. La ventaja que nos proporciona es permitir al usuario examinar el contenido de la base de datos sin tener que recuperar todos los segmentos de vídeo[3][4].

Los servidores de vídeo almacenan los videos digitales fuente y los cuales son referenciados por la base de datos. Los servidores funcionan independientemente del proceso de usuario (sección 0), siendo siempre independientes del modelo de datos y la estructura de índices

implementada en la base de datos.

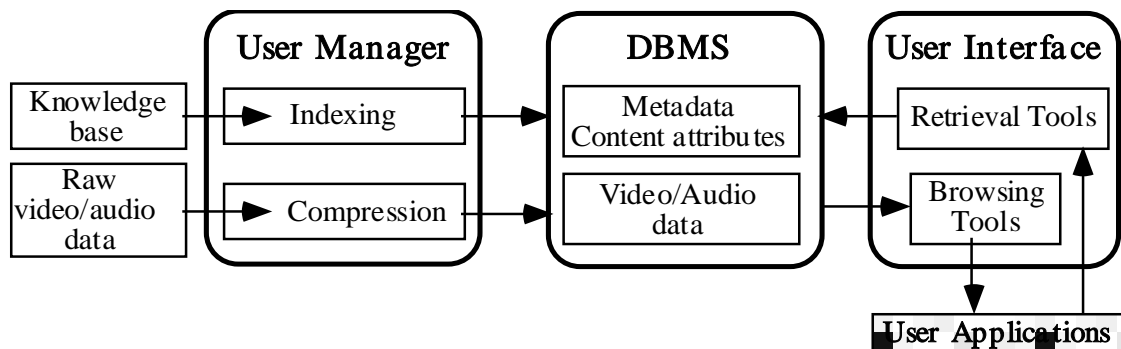


Figura 2: Arquitectura general del sistema

3.2 Módulos del Sistema

3.2.1 Módulo de anotación

El módulo de anotación es un programa cliente (figura 3) que permite la indexación de la semántica (objetos, actividades, contexto) de la fuente vídeo y automáticamente actualiza las tablas de la base de datos. Esta interface ayuda al profesor a añadir anotaciones sobre las secuencias y a actualizar automáticamente la metainformación. El proceso de anotación es un entorno asistido por ordenador, donde el usuario define el inicio-fin de los átomos a indexar por el sistema.

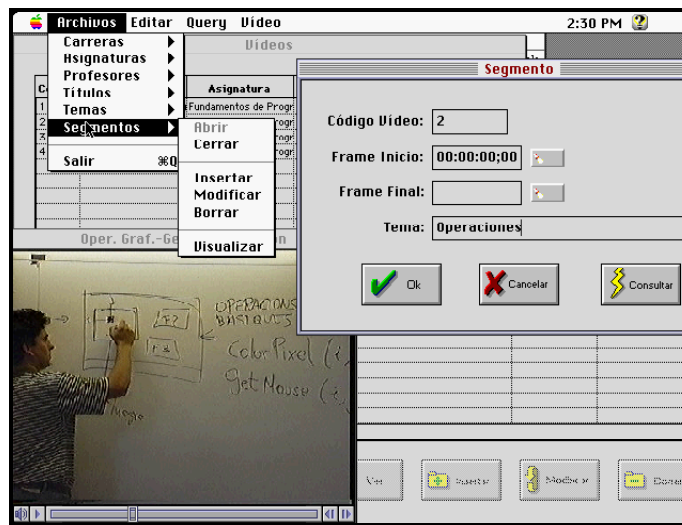


Figura 3: Módulo de Anotación

3.2.2 Módulo de consulta

El módulo de consulta es un programa que permite la interacción de los usuarios (alumnos) con el sistema gestor de la base de datos de vídeo (figura 4). Éste conecta con el SGBD para recuperar la metainformación y posteriormente con los servidores de vídeo para visualizar los segmentos de vídeo deseados. El criterio de recuperación es definido por el usuario interactivamente utilizando las palabras clave existentes y las relaciones definidas en la base de datos.



Figura 4: Módulo de Consulta

El programa muestra todos los atributos y valores asignados de una forma dinámica. Esta forma de construir la consultas asegura que no existan respuestas vacías. La consulta puede ser construida mediante una combinación de operaciones lógicas (AND, OR) y atributos acerca de las clases, carreras, temas, etc. [6][7]. El diseño de la interface se ha desarrollado en dos fases: la primera fue una interface sobre ordenadores Macintosh y implementada en C++. Actualmente estamos generando la misma interface utilizando HTML y *Plug-ins* de Netscape para conseguir un entorno multiplataforma.

3.3 Servidor Tiempo Real

El servidor, implementado para plataformas UNIX Sun Solaris y Silicon Graphics, se divide en dos partes: una de ellas está dedicada a controlar el protocolo de transmisión de vídeo en tiempo real, y la otra a controlar la interacción con la base de datos que devolverá información sobre las secuencias almacenadas. Parte de esta información será la propia dirección del servidor que contiene físicamente los átomos que componen el segmento deseado (de esta forma, aunque la base de datos esté centralizada, los átomos puedan estar almacenados en servidores distribuidos). El protocolo de transmisión en tiempo real es el MovieTalk del QuickTime Conferencing¹ desarrollado por Apple Inc. [15].

3.3.1 QuickTime vs QuickTime Conferencing

El formato de películas QuickTime [15] utiliza la información del recurso que contiene (Movie Resource) para extraer las muestras de cada pista. En dicho recurso se manejan conceptos tales como átomos, tabla de muestras y desplazamiento de *chunks* que permiten extraer las muestras de audio y los *frames* de vídeo de un fichero QuickTime. En cambio, el QuickTime Conferencing sustituye el concepto de pista por el concepto de flujo (*stream*) al estar especialmente diseñado para manejar información en tiempo real. Existe, entonces, una separación conceptual y funcional entre ambos, a pesar de que manejan las mismas estructuras para la descripción de *frames* de vídeo y de muestras de audio. En nuestro caso, nos vemos inexorablemente obligados a confrontar las dos arquitecturas, ya que el servidor UNIX extrae

¹ El QTC es una extensión del sistema del Macintosh desarrollado por Apple Computer [16], con el objetivo de ofrecer servicios de videoconferencias siempre y cuando los conferenciantes estén utilizando una red que disponga de los protocolos TCP/IP, AppleTalk o ISDN.

películas en formato QT pero debe enviarlas en forma de flujos comprensibles para el QTC en el cliente.

3.3.2 El protocolo MovieTalk

A nivel de protocolos de comunicaciones, la arquitectura del sistema presenta la siguiente pila:

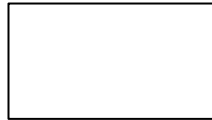


Figura 5 : Arquitectura MovieTalk/TCP-IP

La capa MovieTalk es un protocolo diseñado por Apple Computer, que forma parte del QuickTime Conferencing (está diseñado para que dos ordenadores puedan establecer una videoconferencia sobre una red TCP/IP, AppleTalk o ISDN). Desde el punto de vista del ordenador cliente, el servidor actúa como un conferenciante, y envía al otro extremo (cliente) el *media data* más los datos de control, tal como éste los espera recibir. Esto implica que se deben abrir dos vías de comunicación: datos o MSC (*Media Stream Channel*) sobre UDP y control o CCC (*Conexión Control Channel*) sobre TCP.

3.3.3 Conexión de Control

Permite la interacción y ajuste de ancho de banda entre ambos extremos de la conexión (servidor y cliente). El formato de los mensajes que se intercambian es el siguiente (figura 6):

- Stream Director Component Messages: están relacionados con la negociación de parámetros (formato de compresión de vídeo, audio, dimensión de la ventana de visualización, ...).
- Transport Component Messages: son mensajes que inician la conexión de control o CCC: uno de ellos, es el encargado de indicar al otro extremo de la conexión en que puerto de media data (UDP port) espera que se le envíe el audio y el vídeo. Otro mensaje es el encargado de “transportar” los fragmentos de los frames de vídeo y muestras de sonido (este mensaje no se envía por el CCC sino por el MSC).
- Conference Component Messages: tienen funciones de nomenclatura; el conferenciante que los envía se da a conocer con un determinado nombre.

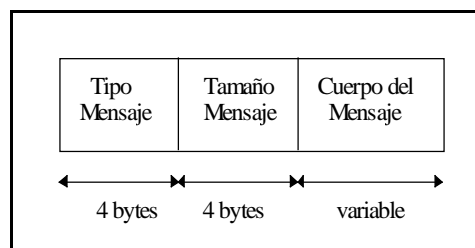


Figura 6: Mensajes de Control

3.3.4 Conexión de Media Data

El enlace de datos en el protocolo MovieTalk se lleva a cabo mediante el *Media Stream*

Channel (MSC) que permite transportar media data (audio y vídeo). En dicho canal se utiliza un servicio no orientado a conexión (protocolo UDP) para transportar los datos, por tanto, no ofrece garantías de entrega segura de los datagramas. Esto es generalmente aceptable para transmisión de vídeo en tiempo real, en donde el objetivo principal es mantener el flujo de información, a costa de la pérdida de algunos fotogramas.

3.3.5 Arquitectura General del Servidor

Por lo explicado anteriormente, para cada cliente debe existir una conexión de control (CCC) y otra de datos (MSC) con el servidor. Existe un tarea o hilo dedicado al CCC y otras dos tareas, una por cada pista del fichero, de cada una de las cuales se deriva un flujo (stream) del MSC (figura 7).

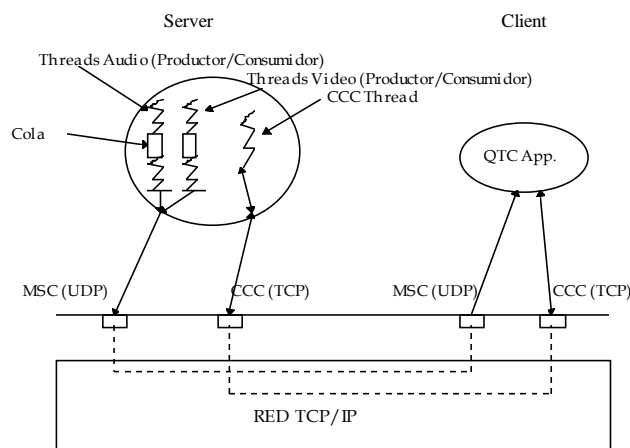


Figura 7: Arquitectura del Servidor de Tiempo Real

3.3.5.1 Control de Flujo

Con la implementación del protocolo MovieTalk del QTC en el servidor no es suficiente para garantizar la mejor QoS (calidad de servicio) de la transmisión en tiempo real. Es necesario, además, diseñar un algoritmo para el control de flujo que permita conservar el ancho de banda, minimizando el efecto 'jitter' debido a la variabilidad de la cadencia en el retardo de los datagramas y el rendimiento de CPU que evite los efectos que producen los diferentes rendimientos de CPU de cada máquina. Para ello, hemos optado por diseñar un algoritmo para el control de flujo con hilos (*threads*) POSIX que permita al servidor UNIX mejorar la QoS [17][18], a partir de los mensajes que envía el QuickTime Conferencing desde el cliente para ajustar el número de frames por segundo que envía el servidor.

3.3.5.2 Uso de threads POSIX

La utilización de procesos típicos de UNIX creados mediante la llamada al sistema 'fork' supone a menudo un consumo innecesario de recursos para el sistema operativo al tener que duplicar todo el espacio de memoria virtual del proceso padre. Para implementar determinados sistemas concurrentes puede resultar más práctico el uso de hilos (*threads*) ya que los métodos de comunicación, por ejemplo, son más simples que con los procesos tradicionales.

El uso de hilos nos ha permitido separar la transmisión de la información en pistas. De esta manera tenemos una tarea dedicada a la transmisión de vídeo y otra a la de audio. Hemos

optado por usar la librería de hilos POSIX para asegurar el portabilidad del código entre SGI y SUN.

3.3.5.3 Colas de Sincronización

Para cada tarea dedicada a la transmisión de una pista hemos implementado un modelo productor/consumidor. De esta manera, tenemos un hilo productor de *chunks* de audio que se encarga de extraer las muestras de audio directamente del fichero de la película y depositarlas en una cola, y un hilo consumidor de audio que las extrae de la cola y las envía. Para la pista de vídeo se utiliza el mismo esquema pero con una cola dedicada exclusivamente a vídeo.

El algoritmo de control de flujo debe asegurar el mejor resultado posible en función del ancho de banda disponible. En las transmisiones de *real-time media* se suele otorgar mayor importancia al audio que al vídeo al considerarse generalmente un servicio prioritario. De esta manera, es conveniente asegurar primero el flujo sobre el canal de audio y posteriormente destinar el resto del ancho de banda a transmitir el mayor número de frames por segundo posible. En el caso del audio, normalmente digitalizado mediante PCM (*Pulse Code Modulation*), partimos de la frecuencia de muestreo para calcular los retardos entre datagramas de audio. El hilo consumidor de audio deberá enviar un datagrama cada vez que se expire el tiempo de espera de este valor del retardo precalculado.

Mientras, la tarea dedicada al vídeo estará en todo momento subordinada al hilo de coordinación (CCC), en éste caso el retardo de envío de los datagramas se calcula en función del *frame-ratio* que soporta el cliente. Esta información vendrá determinada por los mensajes que envía el cliente de QuickTime Conferencing al servidor, en los cuales se informa el número de frames por segundo que el cliente espera recibir para ajustar al máximo el algoritmo de control de flujo del vídeo (figura 8).

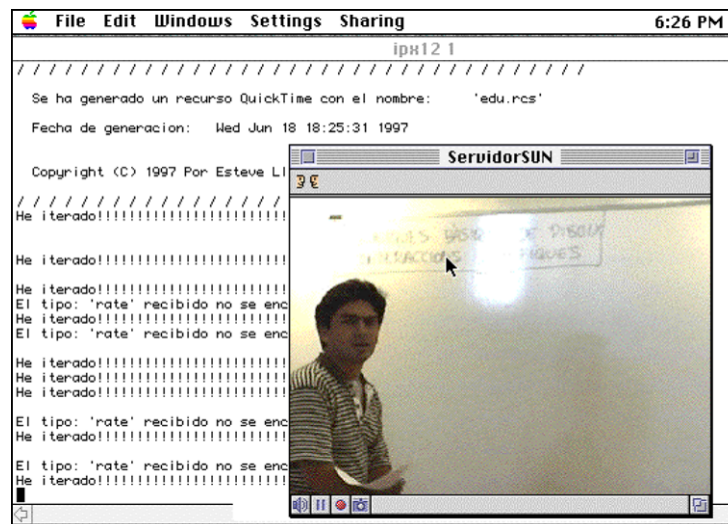


Figura 8: Visualización con Apple Video Conferencing.

4. Conclusiones

En este artículo hemos descrito algunos de los aspectos más importantes de nuestro proyecto sobre transmisión de vídeo digital en tiempo real. Se ha descrito en detalle el diseño del servidor utilizando el protocolo MovieTalk del QuickTime Conferencing, que supone el punto

de partida para resolver el problema de la transmisión del vídeo en tiempo real. Analizando también el problema de control de flujo. Finalmente hemos descrito nuestro algoritmo basado en *threads* como parte de nuestro servidor UNIX para resolver este problema.

Al estar implementado en arquitectura de *threads* optimiza el uso de los procesadores en sistemas de multiprocesamiento. El servidor ha sido probado en una Silicon Graphics Octane con dos procesadores y permite mantener un número elevado de sesiones simultáneas (10-15) sobre una red Ethernet 100Mbs sin degradación de calidad.

Respecto a la indexación, el trabajo extra realizado por el profesor no lo consideramos elevado respecto al proceso de digitalización y compresión del vídeo. Los estudiantes también han mostrado un gran interés en la posibilidad de visualizar a su elección y tantas veces como sea necesario la información audio-visual almacenada.

5. Referencias

- [1] Yoshinobu Tonomura, Akihito Akutsu, Yukinobu Taniguchi, and Gen Suzuki. "Structured Video Computing," *IEEE Multimedia Magazine*, Fall 1994.
- [2] Stephen W. Smoliar, HongJiang Zhang. "Content-Based Video Indexing and Retrieval," *IEEE Multimedia Magazine*, Summer 1994.
- [3] T.D. Little, *et al.* "A Digital On-Demand Video Service Supporting Content-Based Queries," *Proc. ACM Multimedia 93*, Anaheim, CA Aug 1993.
- [4] T.D. Little, Dinesh Venkatesh. "Prospects for Interactive Video-on-Demand," *IEEE Multimedia Magazine*, Fall 1994.
- [5] Samuel DeFazio, Bruce Croft, *et al.* "Integrating IR and RDBMS Using Cooperative Indexing," *SIGIR 95*, Seattle, Washington, USA, July 1995.
- [6] Louis Weitzman, Kent Wittenburg. "Automatic Presentation of Multimedia Documents Using Relational Grammars," *ACM Multimedia 94*, San Francisco, California, 1994.
- [7] Sibel Adali *et al.* "Advanced Video Information System: Data Structures and Query Processing," <http://www.cs.umd.edu/projects/hermes/publications>.
- [8] Antoni Bibiloni, Ricardo Galli, Bartomeu Estrany. "Content Based Retrieval in an Education Video System," to be published in *ED-Media 96*, AACE, Boston, June 1996.
- [9] Antoni Bibiloni. "Representación, Indexación y Recuperación del Contenido Vídeo", *CEIG 95*, Palma de Mallorca, Spain. 1995.
- [10] V.S. Subrahmanian. *Multimedia Databases Systems*, Sushil Jajodia (Eds.) Springer-Verlag 1996.
- [11] S. Gibbs, D. Tsichritzis. *Multimedia programming: Objects, Environments and Frameworks*, ACM Press/Addison Wesley. 1995.
- [12] Kasturi R, Jain R. *Dynamic vision In: Computer Vision: Principles*, IEEE Computer Society Press, Washington, 1991
- [13] Zhang, H.J., Smoliar S.W.. *Developing Power Tools for Video Indexing and Retrieval*, Proc IS&T/SPIE Symp. Electronic Imaging Science and Technology, San Jose, 1994.

- [14] Yoshinobu Tonomura, Akihito Akutsu, Yukinobu Taniguchi, and Gen Suzuki, *Structured Video Computing*, Journal IEEE Multimedia, Fall 1994.
- [15] Apple Computer, "Inside Macintosh: QuickTime Conferencing", Inc. 1995.
- [16] Apple Computer, "Inside Macintosh: QuickTime", Inc. 1992.
- [17] Alan Burns, Andy Wellings, "Real-Time Systems and Programming Languages", 1996.
- [18] Bill O. Gallmeister, "POSIX.4 -- Programming for the real world", 1995.