Universidad de las Ciencias Informáticas
Centro de Geoinformática y Señales Digitales

Método para la generación automática de resúmenes escalables de videos

Tesis presentada en opción al título de Máster en Informática Aplicada

Autor: Ing. Abel Díaz Berenguer
Universidad de las Ciencias Informáticas

Tutor: Dr.C. Yanio Hernandez Heredia
Universidad de las Ciencias Informáticas

Cotutor: Ms.C. Rafael Leodan Cardero Álvarez
Universidad de las Ciencias Informáticas

La Habana
18 de Diciembre de 2014

DECLARACIÓN JURADA DE AUTORÍA

Yo **Abel Díaz Berenguer**, con carné de identidad **85012409324**, declaro que soy el autor principal del resultado que se expone en la tesis para optar por el título de Máster en Informática Aplicada titulada: **Método para la generación automática de resúmenes escalables de videos**.

La investigación se desarrolló en el Centro de Geoinformática y Señales Digitales (GEYSED) de la Facultad 6 de la Universidad de las Ciencias Informáticas (UCI) en el transcurso de los años 2012-2014.

Declaro que todo lo expuesto se ajusta a la verdad, asumo toda responsabilidad moral y jurídica que se derive de este juramento profesional. Además, autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la investigación en su beneficio, así como a los derechos patrimoniales de la misma con carácter exclusivo. Y para que así conste, firmo la presente declaración jurada de autoría, en La Habana a los 18 días del mes de Diciembre del año 2014.

_____

Ing. Abel Díaz Berenguer

Dedicatoria...

A mi niñita Abby y a su maaa, mi esposa Amys, por ser las razones de mi vida y mis fuentes de inspiración para todo lo que emprendo, por su cariño y apoyo, por hacerme inmensamente feliz, porque me hacen hacen sentir que nunca estoy solo y por todo lo que no puedo expresar... Disculpen el tiempo que les he robado estudiando.

A mis padres, Leida y Jose Ramón, por ser eternos paradigmas de mi quehacer cotidiano, por formar mi carácter y mi voluntad para luchar por lo que quiero, por darme un hogar humilde lleno de amor y alegría.

A mi hermana Dagmar y a sus niños Andyno y Andrea, por estar tan lejos y sentirlos tan presentes.

Todo lo que logre en la vida es por ustedes y para ustedes... Los quiero mucho...

Agradecimientos...

A la Universidad de las Ciencias Informáticas y su claustro de profesores por transformarme y formarme, por las oportunidades y apoyo para continuar superándome y por las enseñanzas para la vida.

A mis compañeros de la Facultad 9-6 por su apoyo.

A mis compañeros del Centro de Geoinformática y Señales Digitales por su apoyo y ayuda, aunque estuvieran lejos. En especial a Grethell, Guillermo, Gerdito, Dianita, Pache, Pupo 1, Fer, Jose, Zory, Angelito, Olguita y Frank.

A mis tutores, Rafa y Yanio, por sus revisiones y recomendaciones en el desarrollo del trabajo.

A mis hermanos de la vida, en especial a Neisy y Yadián por su apoyo desde nuestra infancia.

A todas aquellas personas que de alguna forma han contribuido a mi formación como persona y profesional.

Resumen

La recuperación de contenido audiovisual basada en las necesidades de los usuarios, constituye una de las áreas de investigacón más activas y atractivas en la comunidad científica relacionada al procesamiento de imágenes y video. Los métodos desarrollados para facilitar el acceso a estos contenidos se continúan perfeccionando debido a la demanda que poseen, lo que ha implicado el reto de mejorar los ambientes de producción, distribución y recuperación de materiales audiovisuales.

El objetivo de la presente investigación consiste en desarrollar un método para generar automáticamente resúmenes escalables de videos. Se procesan secuencias de video a partir del cómputo de los descriptores de bajo nivel de color y bordes, lo que permitirá realizar la segmentación en tomas del material original, la extracción de los fotogramas principales de estas tomas y su agrupamiento para obtener la representación del contenido visual. Posteriormente se pueden generar resúmenes que se presentan de forma estática o dinámica.

El principal aporte de la investigación radica en que el método propuesto permite procesar secuencias de video que se encuentran en diversas codificaciones. Asimismo, con el método propuesto solamente se debe analizar una vez la secuencia de video original y se pueden generar tantas veces como sea necesario, resúmenes con longitud variable en dependencia de las especificaciones.

Además, en la investigación se establecen los principios para integrar el método en aplicaciones de gestión, procesamiento y transmisión de audiovisuales.

Índice general

Índice de figuras

Índice de tablas

Introducción

En los últimos años se observa un avance tecnológico que se ha evidenciado en el notable aumento de la producción y distribución de contenidos audiovisuales a nivel mundial. Existen estudios que demuestran que los usuarios de internet prefieren consumir, en su mayoría, los archivos audiovisuales disponibles en la red que los transmitidos por la televisión. [Maass y González, 2005, López Vidales y otros, 2011]

Por otra parte, se devela que sitios como YouTube dedicados a compartir materiales audiovisuales en la red, diariamente reciben más de dos millones de visitas [Khan Gramsci, 2011]. Lo anterior, unido al incremento de las capacidades de cómputo y almacenamiento de los sistemas, ha impuesto el reto de perfeccionar los ambientes de producción, distribución y recuperación de contenido audiovisual, pues se predice que continúe aumentando la demanda de estos contenidos [CISCO, 2012]. En este contexto, las técnicas de procesamiento de video se han convertido en una necesidad ante el alto número de contenidos audiovisuales existentes.

Varios investigadores se han dedicado a establecer métodos para describir, procesar, almacenar y recuperar información del contenido audiovisual. Como parte de estos métodos, se pueden observar los que procesan el contenido de un video con la finalidad de generar automáticamente resúmenes del mismo [Truong y Venkatesh, 2007]. El resumen de un video constituye una representación visual sintetizada de la información contenida en el mismo, proporcionando una versión representativa del contenido de la secuencia original. [Zhu y otros, 2004, Truong y Venkatesh, 2007, Valdés y Martínez, 2008, Over y otros, 2008, Xiang-Wei y otros, 2009, Wan y Qin, 2010, Ren y otros, 2010, Emna Fendri, 2010, Parry y otros, 2011, Dan y otros, 2011, Song y otros, 2014]

El acceso al contenido multimedia es una actividad que por lo general implica un proceso de pre-visualización del video por parte del usuario, que debe emplear determinado tiempo en esta tarea. Los métodos para generar resúmenes de video están diseñados para facilitar la navegación sobre una base de datos de video o se pueden utilizar como un producto final, que garantice al usuario acceder rápidamente a posiciones relevantes en la secuencia. [Truong y Venkatesh, 2007]

La mayoría de las aproximaciones que se observan en la literatura para generar resú-

menes automáticos de videos, generan, a partir de una secuencia original, un único resumen [Truong y Venkatesh, 2007, Over y otros, 2008, Parry y otros, 2011, Dan y otros, 2011, Song y otros, 2014]. Este tipo de resumen, según las consideraciones del autor de esta investigación, a veces puede ser insuficiente. En ocasiones es deseable lograr una presentación personalizada para cada usuario y los resúmenes escalables resultan útiles para hacer frente a la diversidad de preferencias y lograr mayor usabilidad.

La creación de métodos, que permitan obtener resúmenes de video escalables, ha despertado interés para investigadores de la temática durante varios años, lo que se evidencia en los trabajos de [Zhu y otros, 2003, Herranz y Martínez, 2010, Herranz Arribas, 2010, Herranz y otros, 2012, Díaz Berenguer, 2014]. Se asume por resumen escalable, aquel en el que la sinopsis se puede adaptar a determinadas preferencias o condiciones; por ejemplo, la resolución, la tasa de bits por segundo, la longitud o duración. Para lograr este objetivo, un procedimiento que genere resúmenes escalables debe garantizar que, una vez procesada la secuencia de video original, exista la posibilidad de generar varias síntesis dependiendo de las condiciones o preferencias establecidas. [Herranz Arribas, 2010]

La presente investigación tiene sus orígenes en el desarrollo de un conjunto de aplicaciones para la gestión, procesamiento y transmisión de audiovisuales en el centro GEYSED de la UCI. Después de desplegadas estas aplicaciones: Plataforma de Televisión Informativa (PRIMICIA) [Hernández García y otros, 2010], Sistema de Gestión, Procesamiento y Transmisión de Contenidos Audiovisuales (SIAV) [Díaz Berenguer y otros, 2013] y Plataforma de Gestión, Catalogación y Publicación Web de Contenidos Audiovisuales (AGORAV) [Rodríguez y otros, 2014], su explotación ha demostrado que se caracterizan por el constante aumento de la cantidad y diversidad de materiales audiovisuales, gestionados y posteriormente almacenados para su utilización. Además se evidencia la utilización de varias codificaciones de video, dado fundamentalmente por los requisitos establecidos por los clientes potenciales. En las aplicaciones antes mencionadas, se debe garantizar el proceso de catalogación de los materiales audiovisuales gestionados para facilitar su búsqueda y utilización. En éstas se han desarrollado metodologías y algoritmos [Hernández Heredia, 2010, Díaz Ales y Alonso Guerrero, 2012], que contribuyen a lograr la catalogación automática o semiautomática, pero aún es necesario la anotación manual de determinados datos.

Durante el proceso de catalogación de los materiales se presentan las siguientes situaciones:

1. Los usuarios que realizan la catalogación obtienen alguna información referida al material de determinada fuente, sin embargo, en ocasiones se requiere adquirir al menos la idea principal del contenido audiovisual mediante la visualización de varios fragmentos o de la secuencia completa.

2. Los usuarios que realizan la catalogación no poseen, ni logran obtener de ninguna fuente, información referida al material a catalogar, lo que implica la visualización del mismo para su correcta descripción.

En estas tres aplicaciones se notan insuficiencias en el proceso de catalogación, debido a que los usuarios actualmente obtienen las perspectivas del contenido audiovisual navegando, en ocasiones aleatoriamente, sobre la secuencia de video.
Adicionalmente, en AGORAV se persigue el objetivo de proveer a los usuarios de materiales audiovisuales disponibles en la web para consumir bajo demanda. Es común notar en los consumidores videos, generalmente no de corta duración, que antes de proceder a la visualización completa del material, realizan una pre-visualización del mismo en aras de obtener una idea de su contenido y posteriormente visualizarlo completamente, si lo consideran de interés. Además aunque actualmente no se encuentra en explotación, para esta aplicación está previsto un módulo en el que los usuarios pueden compartir videos, lo que brindará la posibilidad de que sean publicados por éste. Ello traería consigo que en determinados entornos, sea necesario realizar una revisión de los videos para aprobar su publicación. En las condiciones actuales, la persona que se encargaría de la revisión debería visualizar o pre-visualizar la secuencia de video, para certificar su contenido y determinar si es posible su publicación.
Lo expuesto anteriormente revela limitaciones que determinan el siguiente **problema de investigación**: el contenido de un video, determinado mediante la navegación sobre la secuencia completa o pre-visualizando determinados fragmentos, dificulta el acceso a posiciones relevantes para la descripción y toma de decisiones sobre el mismo.
El **objeto de estudio** en el que se enfoca la investigación está constituido por: los métodos para obtener resúmenes de videos.
Con el propósito de solucionar el problema de planteado, se establece como **objetivo general** de la investigación: desarrollar un método que permita generar resúmenes escalables de videos, facilitando el acceso a posiciones relevantes para la descripción y toma de decisiones sobre el

mismo.

Los **objetivos específicos** son:

- Determinar las etapas y fases necesarias para obtener resúmenes escalables de una secuencia de video.
- Diseñar un método para obtener resúmenes escalables de una secuencia de video.
- Validar el método propuesto para obtener resúmenes escalables de una secuencia de video.

Una vez que se han planteado el problema de investigación, el objeto de estudio y los objetivos, se define como **campo de acción** de la investigación: los métodos para generar resúmenes escalables de video.

La **hipótesis de trabajo** que guía la presente investigación es que: si se desarrolla un método para generar automáticamente resúmenes escalables de un video, se facilitará el acceso a posiciones relevantes para la descripción y toma de decisiones sobre el mismo.

Para el desarrollo de la investigación se emplearon **métodos teóricos y empíricos**.

- El método **hipotético-deductivo** para elaborar la hipótesis de investigación y validarla.
- El método **histórico-lógico** para estudiar la trayectoria y evolución histórica de los trabajos anteriores para generar resúmenes escalables de video, que constituyen referente teórico-prácticos, así como establecer las tendencias generales que rigen el funcionamiento y desarrollo de los mismos.
- El **analítico-sintético** con el objetivo de descomponer el objeto de estudio de la investigación y procesar los fundamentos científicos y las teorías relacionadas, lo que permite extraer los aspectos significativos que sustentan el método propuesto.
- La **modelación** para el diseño del método propuesto en la investigación.
- La **encuesta** se realiza para validar la escalabiliad y usabilidad del método propuesto.
- La **experimentación** se utiliza para validar el método propuesto, en condiciones controladas, a partir de un conjunto de videos de prueba.

El **aporte** de la investigación descrita en este documento, consiste en un método en el que se establecen las etapas y fases necesarias para obtener resúmenes escalables de un video, teniendo en cuenta la combinación de descriptores de bajo nivel. El método propuesto permite generar resúmenes para distintas codificaciones de video. Por otra parte el resultado posee valor práctico, determinado por mejoras en las tareas del capital humano destinado a actividades de recuperación y descripción de archivos audiovisuales. Asimismo la investigación establece los elementos necesarios para la inclusión del método en aplicaciones de gestión, procesamiento y transmisión de archivos audiovisuales que se han desarrollado en la UCI, específicamente en GEYSED.

Estructura del documento

El presente documento se encuentra dividido en tres capítulos. En el primer capítulo se exponen los fundamentos teóricos asociados al procesamiento de videos para la obtención de resúmenes, asimismo se realiza un análisis de aproximaciones relacionadas estudiadas en la literatura. En el segundo capítulo se expone detalladamente el método propuesto para la obtención de resúmenes escalables de video. Por su parte, en el tercer capítulo, se evidencian un conjunto de pruebas experimentales realizadas para validar el método propuesto, además se establecen los elementos esenciales para la inclusión del método en aplicaciones de gestión, procesamiento y transmisión de contenidos audiovisuales desarrolladas en GEYSED. Se incluyen las conclusiones y recomendaciones derivadas de la investigación, así como las referencias bibliográficas utilizadas. El documento también contiene un conjunto de acrónimos, siglas y anexos para facilitar la comprensión de la investigación.

Capítulo 1

Fundamentos teóricos de la investigación

Con el objetivo de facilitar la comprensión del alcance de la investigación en el presente capítulo se exponen los fundamentos teóricos asociados al dominio del problema. Se aborda la estructura de los videos y la descripción de su contenido. Por otra parte se estudian las técnicas para segmentar físicamente el contenido de los videos, las medidas de similaridad y los métodos de agrupamiento. Posteriormente se analizan los resúmenes de video y sus modalidades, las características de los resúmenes escalables, así como aproximaciones existentes para dar solución al problema que se ha planteado en la investigación.

1.1. Estructura del contenido de un video

Las técnicas de análisis sintáctico sirven para extraer la estructura del contenido de un video. Estas técnicas se aplican en dependencia de la modalidad del video y existen dos posibles tipologías. [Cózar, 2010]

- Una de las modalidades es la que posee el contenido basado en un guión, el video tiene una estructura bien definida. Ejemplos de este tipo de videos lo constituyen los seriales, las películas, noticias y documentales.
- La otra modalidad de video es aquella en la que su contenido no se basa en un guión, porque los eventos que ocurren son aleatorios. El ejemplo clásico de este tipo de videos, lo constituyen las secuencias resultantes de grabaciones realizadas con cámaras de videovigilancia.

Es posible también observar una tipología mixta que considera casos intermedios, secuencias que poseen segmentos predecibles y segmentos totalmente aleatorios. Se puede ejemplificar con las secuencias generadas en las transmisiones de determinados deportes. [Cózar, 2010]

Para la investigación, se trabaja con el video cuyo contenido se basa mayoritariamente en un guión, por lo que se puede definir el video como una estructura jerárquica en la que se identifican claramente los componentes: escenas, tomas y fotogramas, ver figura 1.1:

- La escena representa un concepto de alto nivel. Normalmente está constituida por la sucesión de tomas adyacentes, aunque puede existir alguna escena que esté constituida por una sola toma. Lo que determina que sea una escena es que se aborde un tema o asunto de forma coherente, ya que representan las mínimas sub-secuencias con significado completo. Las tomas que pertenecen a una escena, poseen relación semántica y están cerca temporalmente. La escena se puede considerar como una frontera semántica en el video. [Sáez Peña, 2006, PRASANNA, 2013]

- Una toma es un segmento de una secuencia de video que no posee cortes ni transiciones entre los cuadros que la conforman, es decir, una secuencia de cuadros consecutivos resultantes de una operación continua de grabación. Debido a esta condición, la toma posee un fondo consistente característico. Por el contrario de la escena, una toma es considerada un límite físico en el video. [Sáez Peña, 2006, PRASANNA, 2013]

- Los fotogramas o cuadros constituyen la unidad básica del video. Un fotograma no es más que una imagen. Una colección de fotogramas da lugar a una toma y en esta colección siempre existe un fotograma principal, que es el más representativo o destacado de la colección, incluso, en tomas muy largas se pueden considerar más de un fotograma principal. [Sáez Peña, 2006, Cózar, 2010, PRASANNA, 2013]

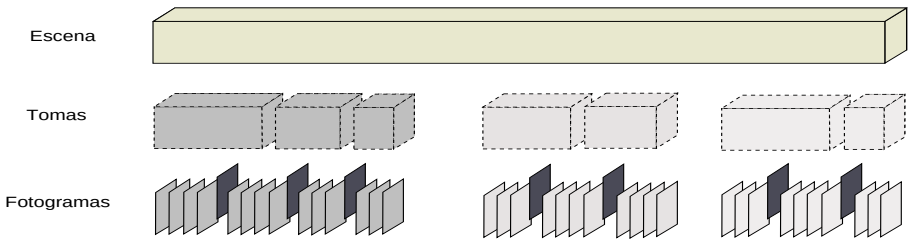

Figura 1.1: Estructura jerárquica de una secuencia de video basada en contenido, fuente [Mendi y Bayrak, 2010].

1.2. Descripción del contenido de un video

Comprender la forma en que se representa la información audiovisual de un video, implica ofrecer la definición y caracterización de los descriptores audiovisuales. Para la definición, se toma la propuesta del Diccionario de la Lengua Española [DRA, 2001] de descriptor *"constituye un término o símbolo válido y formalizado que se emplea para representar inequívocamente los conceptos de un documento o de una búsqueda"*.

Al aplicar esta propuesta en el dominio del procesamiento de imagen y video, se puede establecer por descriptor toda forma de representación de los atributos o características de una imagen y por consiguiente de un video.

En investigaciones precedentes se plantea que los descriptores de archivos audiovisuales surgen como respuesta a la necesidad de caracterizar la información audiovisual de forma objetiva y automatizada [Díaz Espinoza, 2011, Boullosa García, 2011]. Mantienen información relevante y generalmente se expresan mediante vectores sobre algún espacio matemático particular. [Hernandez Heredia y otros, 2012]

Al mismo tiempo [Díaz Espinoza, 2011, Boullosa García, 2011, Hernandez Heredia y otros, 2012], clasifican los diferentes tipos de descriptores de imagen o video atendiendo al grado de abstracción del contenido que representan, siendo posible agruparlos en dos categorías principales:

- Descriptores de bajo nivel o de información general: extraen la información más básica del material audiovisual. Proporcionan una descripción respecto a características o atributos visuales como el color, las formas, regiones, texturas o los movimientos presentes en la imagen. Los algoritmos para el cómputo de estos descriptores funcionan con un alto grado de fiabilidad. [Hernandez Heredia, 2013]

  Por la importancia de estos descriptores para la investigación, se hará referencia a dos características de la imagen que se logran representar mediante los mismos y se utilizarán en la propuesta de solución en el capítulo 2.

  - Color: constituye la cualidad más básica del contenido visual. Estos descriptores permiten describir el color mediante sus propiedades representando su distribución. Para el trabajo con el color y lograr una correcta caracterización es de su-

ma importancia tener en cuenta el espacio de color de la imagen que se analiza. [Swain y Ballard, 1991, Sáez Peña, 2006]

- Forma: es una característica con la propiedad de brindar información semántica, se basa en la capacidad del ojo para reconocer los objetos solamente observando su forma. La base para extraer esta información es realizar una segmentación de la imagen de forma similar a como la realiza el sistema visual humano. Existen una serie de algoritmos que permiten una buena aproximación a la forma de un objeto. Estos descriptores permiten representar las regiones, contornos y formas. [Kim y Hwang, 2002, Veeraraghavan y otros, 2005, Sáez Peña, 2006, Bosch y otros, 2007]

- Descriptores de contenido de alto nivel o de información de dominio específico: conocidos en el ámbito del procesamiento audiovisual como los descriptores semánticos debido a su capacidad para proporcionar información acerca de los objetos, acciones o eventos que constituyen una escena. Estos descriptores logran la caracterización semántica utilizando los de bajo nivel para obtener sus características visuales y referir las diferentes categorías semánticas. Su utilización permite representar directamente el contenido de una imagen o video. Este grupo de descriptores son considerados de alto nivel, deben ser desarrollados por un humano que garantice la inteligencia y precisión de su funcionamiento. [Hernandez Heredia, 2013]

Dadas las categorías anteriores como parte de los descriptores de bajo nivel, es posible establecer otra clasificación de descriptores según su nivel de aplicación sobre las regiones de la imagen.

- Descriptores locales: se definen como locales debido a que inciden sobre determinadas regiones de interés, previamente establecidas o identificadas. Permiten obtener un vector de características de la región en cuestión y puede tenerse en cuenta la información contenida, tanto en la región de interés como en la regiones adyacentes y en su vecindario. Por lo general estas regiones son caracterizadas por puntos de interés o puntos destacados y pueden referirse a bordes o secciones específicas de la imagen. El descriptor finalmente se constituye por la totalidad de los vectores de características calculados. Se logra la descripción de las imágenes mediante vectores que representan sus puntos o regiones características. [Tuytelaars y Mikolajczyk, 2008]

- Descriptores globales: precisamente constituyen lo opuesto a los locales, representan el contenido de la imagen en un único vector o matriz de características. Estos descriptores son atractivos en el campo, ya que poseen la capacidad de encapsular gran cantidad de información de la imagen, requiriendo una pequeña cantidad de datos para describirla. En la práctica, son descripciones que resumen o engloban todas las características de la imagen en un vector único. Los descriptores globales se pueden considerar simples porque existen varias implementaciones para lograr su extracción. A pesar de su simplicidad, este tipo de descriptores es ampliamente utilizado para diferentes aplicaciones debido, entre otras cosas, a su bajo coste computacional unido a que ofrecen una precisión bastante aceptable. [Díaz Espinoza, 2011, Boullosa García, 2011]

También se pueden clasificar los descriptores por la dimensión en la que se enfocan. [Laptev, 2005, Bregonzio y otros, 2009, Hernandez Heredia y otros, 2012, Hernandez Heredia, 2013]

- Espaciales: se aplican las funciones para la extracción de características a cada uno de los fotogramas por separado, tratando los videos como un conjunto de imágenes.
- Temporales: se utilizan para tener en cuenta la información temporal, extraen características utilizando la variable temporal del video. Para esto se debe dar un seguimiento temporal de la información obtenida con un descriptor local. El objetivo final es lograr la representación de las características considerando varios fotogramas.
- Espacio-Temporales: en esta categoría se agrupa la combinación de los dos anteriores.

Finalmente y muy asociado a las taxonomías basadas en la dimensión en que se enfocan, se definen los descriptores en un video de acuerdo a la estrategia que se sigue para escoger los fotogramas sobre los que se aplicará la extracción de características. [Díaz Espinoza, 2011, Boullosa García, 2011]

- Basado en fotogramas principales: agrupa las estrategias que extraen solamente las caracteríticas de los fotogramas principales del video.
- Grupo de fotogramas: es una estrategia que extrae las características de un grupo o grupos adyacentes de fotogramas del video.

- Fotograma por fotograma: se tienen en cuenta todos los fotogramas del video para extraer sus características y ser analizados.

En esta memoria, el autor utiliza el término vector de características, para referirse a la descripción del contenido audiovisual de los fotogramas y por consiguiente del video. En las definiciones 1 y 2 se ajustan las propuestas de [Sáez Peña, 2006] para la investigación.

DEFINICIÓN 1 *El vector de características de una imagen, está constituido por el conjunto de aspectos que la define. Permite representar la información audiovisual de la imagen.*

DEFINICIÓN 2 *Una función de distancia o similitud permite medir las semejanzas existentes entre dos imágenes a través de su vector de características.*

Los descriptores de interés para la representación de la información audiovisual en esta investigación, son los descriptores de bajo nivel globales y locales. Se extraen las características de forma espacial mediante una estrategia que recorre fotograma por fotograma.
En tesis doctoral defendida recientemente [Hernandez Heredia, 2013] y publicación asociada [Hernandez Heredia y otros, 2012] se establecen las propiedades a considerar para seleccionar adecuadamente un descriptor, las mismas se asumen para la investigación y se relacionan seguidamente:

- Simplicidad: *"el descriptor debe representar las características extraídas de la imagen de manera clara y sencilla para permitir una fácil interpretación de su contenido".*
- Repetibilidad: *"el descriptor generado a partir de una imagen debe ser independiente del momento en el que se genere".*
- Diferenciabilidad: *"dada una imagen, el descriptor generado debe poseer alto grado de discriminación respecto a otras imágenes y al mismo tiempo contener información que permita establecer una relación entre imágenes similares".*
- Invarianza: *"cuando existen deformaciones en la representación de dos imágenes, es deseable que los descriptores que las representan aporten la robustez necesaria para poder relacionarlas, aún bajo diferentes transformaciones".*

1.2.1. Histogramas de color

En el dominio de procesamiento audiovisual es muy común el término histograma. Los histogramas permiten representar las características de objetos contenidos en una imagen o regiones de esta, si se asume como descriptor local. Asimismo, se puede utilizar para representar propiamente un aspecto de la imagen, asumiéndose como descriptor global. Un histograma es una colección de conteos o estadísticas de algún dato, organizada en un grupo predefinido de divisiones, denominadas *bins* en la literatura anglosajona [Bradski y Kaehler, 2008]. Los histogramas de color han ganado gran popularidad [Don y Uma, 2009, Qifan y otros, 2013, Zhang y Wang, 2012, Ejaz y otros, 2012, Mohanty y Kanungo, 2013, Liu y Yang, 2013] en el procesamiento audiovisual por su capacidad para representar la distribución de color de objetos o imágenes.
Asumiendo los criterios de [Don y Uma, 2009, Qifan y otros, 2013, Zhang y Wang, 2012, Ejaz y otros, 2012, Liu y Yang, 2013] como se verá en el capítulo 2 los histogramas de color constituyen uno de los descriptores de bajo nivel seleccionados en esta investigación para caracterizar la información audiovisual de los fotogramas que constituyen el video.

1.2.2. Contornos

Los contornos de una imagen permiten representar la forma de los objetos que aparecen en la misma y se pueden determinar mediante varias aproximaciones. Por ejemplo, a través de los bordes, se pueden representar cambios significativos de intensidad local, lo que facilita caracterizar la forma de los objetos, separar regiones o identificar cambios de iluminación. La detección de bordes está considerada como una de las operaciones fundamentales en el dominio de la visión por computadoras y existen diversas aproximaciones para su cómputo, basado en la detección mediante transformaciones sobre la imagen para lograr una representación alternativa de todos sus datos. [Chandrakar y Bhonsle, 2012]
Los métodos para detectar contornos referenciados en la literatura poseen un comportamiento muy similar [Bradski y Kaehler, 2008, Chandrakar y Bhonsle, 2012]. Sin embargo varios autores [Boyle y Thomas, 1988, McIlhagga, 2011, Ali y Clausi, 2001, Panetta y otros, 2011, Davies, 2012, McIlhagga, 2011, Panetta y otros, 2011] refieren que el algoritmo ideado por John Canny 1986 [Canny, 1986] permite obtener resultados satisfactorios, incluso cuando la imagen posee ruido, por lo cual se considera un buen detector [Chandrakar y Bhonsle, 2012].

En la investigación se selecciona la propuesta de John Canny [Canny, 1986] como descriptor de bajo nivel, para caracterizar los bordes presentes en los fotogramas del video. Para su selección se asumen los argumentos expuestos por dicho autor [Canny, 1986], que se citan en otros estudios [Boyle y Thomas, 1988, González y Woods, 2002, Ding y Goshtasby, 2001, Acharya y Ray, 2005]:

1. Baja tasa de error, minimiza la detección de bordes inexistentes.
2. Garantiza buena precisión para la localización del borde.
3. Solamente ofrece un resultado por borde.

1.3. Determinación de los límites entre tomas

En las aplicaciones de procesamiento audiovisual la detección de tomas constituye un proceso automático, que persigue el objetivo de detectar los límites de cada toma [Sáez Peña, 2006]. Dicha temática ha captado la atención de varios investigadores debido a que se considera un paso necesario para el análisis, la indexación, creación de resúmenes, búsqueda u otras operaciones basadas en el contenido de un video. [Lupatini y otros, 1998, Bescós y otros, 2005, Sáez Peña, 2006, Don y Uma, 2009, Yinzi, 2010, Mohanta y otros, 2010, Smeaton y otros, 2010, Mendi y Bayrak, 2010, Amiri y otros, 2011, Lee y otros, 2011a, Xiang y otros, 2011, Zhang y Wang, 2012, Chavan y otros, 2013, Lu y Shi, 2013, Zhe-Ming y Yong, 2013, Jiang y otros, 2013, Qifan y otros, 2013, Thounaojam y otros, 2014, Mohanty y Kanungo, 2013] En la variedad de aproximaciones para la segmentación de videos, se evidencian tres aspectos fundamentales que permiten su diferenciación [Smeaton y otros, 2010, Xiang y otros, 2011, Zhang y Wang, 2012, Mohanty y Kanungo, 2013] que se tienen en cuenta para la propuesta.

- Métodos de representación del contenido visual de los fotogramas.
- Las medidas de continuidad o similaridad entre los fotogramas.
- Los modelos de clasificación o detección.

Según los resultados actuales se puede establecer un método capaz de determinar diferencias significativas entre los efectos de edición que representan cortes. En este

punto, los métodos empleados aún no logran establecer con exactitud la presencia de un cambio de toma abrupto o gradual [Sáez Peña, 2006, Smeaton y otros, 2010]. Esta es una temática que continúa en estudio, observándose diferentes aproximaciones, [Don y Uma, 2009, Yinzi, 2010, Mohanta y otros, 2010, Smeaton y otros, 2010, Mendi y Bayrak, 2010, Amiri y otros, 2011, Lee y otros, 2011a, Xiang y otros, 2011, Zhang y Wang, 2012, Chavan y otros, 2013, Lu y Shi, 2013, Zhe-Ming y Yong, 2013, Jiang y otros, 2013, Qifan y otros, 2013, Mohanty y Kanungo, 2013, Thounaojam y otros, 2014] pero a consideración del autor de la investigación la efectividad de los resultados es similar.

Para realizar el proceso de segmentación, se debe considerar la similaridad entre fotogramas, en este caso teniendo en cuenta aquellos que son diferentes. Los valores de similaridad cuando se buscan los cambios de tomas abruptos, se determinan mediante funciones de similitud o distancia aplicadas a los fotogramas contiguos, permitiendo establecer sin ambigüedades cuándo hay presencia de un límite y cuándo no [Bescós y otros, 2005, Sáez Peña, 2006, Jinhui y otros, 2007, Smeaton y otros, 2010, Chavan y otros, 2013]. Las funciones de distancia se computan a partir de los valores que toman los descriptores de los fotogramas del video. Por otra parte, en el caso de los cambios de tomas graduales, también se debe tener en cuenta la similaridad, pero no entre fotogramas contiguos o adyacentes. Dicha estrategia no brinda información relevante, pues para estos casos existirá ambigüedad. Por este motivo se deben establecer otras estrategias que permitan la identificación de las tomas graduales.

En investigaciones precedentes [Bescós y otros, 2005, Sáez Peña, 2006] se enuncian definiciones que se deben tener en cuenta para el proceso. Seguidamente se establecen ajustadas a la propuesta de solución.

DEFINICIÓN 3 *Sea un fotograma i, que pertenece a una secuencia de video S, representado por un vector de características v_i y una función de distancia f_d definida; la distancia entre fotogramas contiguos o adyacentes $d_i; d_{i-1}$ se determina evaluando f_d para v_i y v_{i-1}.*

Asimismo se puede enunciar una definición similar para una función de similitud.

DEFINICIÓN 4 *Sea un fotograma i, que pertenece a una secuencia de video S, representado por un vector de características v_i y una función de similitud f_s definida, el valor de similitud entre fotogramas contiguos o adyacentes $s_i; s_{i-1}$ se determina evaluando f_s para v_i y v_{i-1}.*

Las funciones referidas en las definiciones, 3 y 4, se pueden utilizar para comparar las semejanzas existentes entre dos fotogramas. En la propuesta se hará alusión a estas funciones teniendo en cuenta que deben estar normalizadas entre cero y uno. Aunque para propósitos prácticos se pueden considerar funciones iguales, la diferencia radica en que se expresará la mayor semejanza cuando una función de similitud $f_s \approx 1$ y una función de distancia $f_d \approx 0$.

En la investigación para la segmentación de tomas se emplearán dos métodos fundamentales, uno basado en los histogramas de color de los fotogramas del video. Este ha sido un método utilizado por varios años para la segmentación de tomas y con probada efectividad para la detección de cambios de tomas abruptos o cortes. Debido a que este método no es efectivo para la detección de cambios de tomas graduales, se emplea además, otro basado en los bordes presentes en los fotogramas, mediante una estrategia de ventana deslizante.

1.3.1. Funciones de similaridad

Las funciones de similaridad permiten determinar el nivel de semejanza entre elementos pertenecientes a un mismo dominio. Las mismas se utilizan en diversas aplicaciones y tópicos de investigación. El estudio realizado y propuesto en [Deza y Deza, 2012] constituye una fuente teórica, que a consideración del autor, detalla las características de estas funciones y refiere cuáles son las más utilizadas.

En el procesamiento digital de imágenes y señales de video se puede observar la utilización de las funciones de distancia y similitud, para resolver problemas de procesamiento audiovisual relacionados con la detección y clasificación de objetos, la segmentación y el reconocimiento de patrones. [Basseville, 1989, Rubner y otros, 2001]

Las funciones de distancia y similitud deben satisfacer las condiciones que se relacionan a continuación en la definiciones 5 y 6 [Cózar, 2010].

DEFINICIÓN 5 *Sean i, j dos elementos en igual modelo al computar la función de distancia f_d debe cumplirse:*

- $f_d(i, j) \geq 0$.
- $f_d(i, i) = 0$.
- $f_d(i, j) = f_d(j, i)$.

Además, para considerarla un distancia métrica debe cumplirse:

- $f_d(i,j) \leq f_d(i,k) + f_d(k,j)$

DEFINICIÓN 6 *Sean i, j dos elementos en igual modelo al computar la función de similitud f_s debe cumplirse:*

- $f_s(i,j) \leq 1$.
- $f_s(i,i) = 1$
- $f_s(i,j) = f_s(j,i)$.

En la solución que se explicará en el capítulo 2, se identificará la utilización de la función de distancia Euclidiana y la función de similitud Coseno [Deza y Deza, 2012].

1.3.2. Definición del fotograma principal

Para este proceso se pueden aplicar criterios de otras aproximaciones que establecen como fotograma principal el primero, el último o un fotograma de la toma seleccionado de forma aleatoria. Esto no se puede aplicar la solución, porque no es posible garantizar que este es el fotograma representativo. Otros procedimientos comparan los fotogramas adyacentes utilizando descriptores bajo nivel como el color o la textura. Consideran aquel fotograma con más información como fotograma principal, pero no necesariamente la cantidad de información es una medida representativa. Otras aproximaciones tienen en cuenta la estimación de la cantidad de movimiento en la toma, determinando el fotograma con la menor cantidad de movimiento local como principal. Aplicar esta variante implica la dificultad de determinar la cantidad de movimiento. [Lux y otros, 2007, Liping y otros, 2010, Borth y otros, 2008, Hampapur y otros, 2012, Chen y otros, 2011, Ejaz y otros, 2012]

Por su parte, otros investigadores simplemente agrupan en forma particional los fotogramas de la toma y el fotograma más cercano al centro del grupo se asume como más representativo [Amiri y otros, 2011]. En esta variante se puede utilizar la similaridad aunque para aplicarla no es necesario agrupar. Se puede determinar el fotograma más representativo como el que más se asemeje al resto combinando con la variante asociada a la mayor cantidad de información aportada.

1.4. Métodos de agrupamiento

El término agrupamiento es muy utilizado en los estudios de procesamiento estadístico, en las ciencias de la computación y en las aplicaciones de procesamiento audiovisual. [Jain y otros, 1999, Pérez Suárez y otros, 2008, Dumont y Mérialdo, 2008, Amiri y otros, 2011, Chan y otros, 2011, Song y otros, 2014]

En [Xu y C. Wunsh, 2009] se manifiesta que universalmente no existe un consenso preciso en cuanto a la definición del término agrupamiento. Por su parte [Everitt y otros, 2001], indica que *"...ofrecer una definición formal de clúster es difícil, porque incluso puede estar fuera de lugar..."*. Basado en los criterios de [Jain y otros, 1999, Arco García, 2008, Hastie y otros, 2009, Witten y otros, 2011] en la investigación se considera que el agrupamiento es la división de cierta cantidad de datos en determinado número de conglomerados (grupos, subgrupos o categorías). A su vez, un conglomerado está constituido por un conjunto de datos que son iguales o muy similares. Los conglomerados se conforman o dividen siguiendo un criterio de semejanza.

Esta definición de agrupamiento se puede describir en términos de homogeneidad interna y separación externa [Arco García, 2008, Xu y C. Wunsh, 2009] estableciendo que los datos pertenecientes al mismo grupo deben ser iguales o similares entre sí, mientras que los datos pertenecientes a grupos distintos deben ser diferentes.

Las técnicas de agrupamiento se clasifican en particionales y jerárquicas [Arco García, 2008, Xu y C. Wunsh, 2009, Pérez Suárez y otros, 2008, Everitt y otros, 2001]. El agrupamiento particional divide directamente los datos en un número específico de grupos sin tener en cuenta una estructura jerárquica. Por otra parte, el agrupamiento jerárquico conforma una secuencia de particiones anidadas, ya sea desde grupos aislados que incluyen a todos los grupos o viceversa. El primer caso es conocido como agrupamiento jerárquico aglomerativo y el otro caso es denominado agrupamiento jerárquico divisivo.[Arco García, 2008, Xu y C. Wunsh, 2009, Pérez Suárez y otros, 2008, Everitt y otros, 2001]

Ambas técnicas de agrupamiento jerárquico, aglomerativa y divisiva, pretenden la organización de los datos en una estructura jerárquica basándose en una matriz de proximidad. Los resultados del agrupamiento jerárquico se representan frecuentemente por un árbol binario o dendrograma, según se ilustra en la figura 1.2. El nodo raíz del dendrograma representa el grupo que aglomera todo el conjunto de datos y en cada nodo hoja se ubica un elemento que está agrupado según

la relación de su predecesor. Los nodos intermedios, por lo tanto, describen el grado en que los grupos son próximos uno al otro. La altura del dendrograma expresa la distancia entre cada par de grupos. Se pueden obtener distintos resultados de agrupamiento mediante la reducción en diferentes niveles y dendrogramas. Esta representación proporciona descripciones informativas y una visualización de las posibles estructuras de agrupamiento de datos, especialmente cuando existen relaciones jerárquicas reales en los datos [Xu y C. Wunsh, 2009, Everitt y otros, 2001].

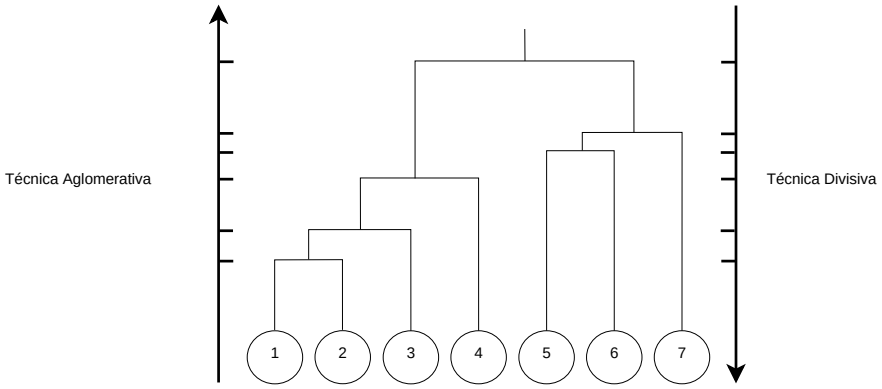

Figura 1.2: Dendograma resultante de un agrupamiento jerárquico, fuente [Xu y C. Wunsh, 2009]

1.5. Resúmenes de video

La creación de resúmenes de video involucra diversos niveles de complejidad, como se evidencia en los resultados de estudios realizados previamente. En [Valdés y Martínez, 2008, Herranz y Martínez, 2010, Sasonkgo, 2011] se muestra el modelo genérico para la conformación de un resumen de video y se evidencian dos procedimientos fundamentales: el análisis y la generación, que se tendrán en cuenta para la solución propuesta en el capítulo 2.

Durante el análisis se realizan todas las operaciones destinadas a extraer la información del contenido original, con independencia del uso posterior de dicha información. Se desarrolla la detección o segmentación de las unidades básicas de procesamiento del resumen. Las unidades de procesamiento pueden ser los fotogramas, las tomas o las escenas. Posteriormente se realiza la extracción y análisis de las características, en dependencia los valores que toman los descriptores en dichas unidades. Estas características se tendrán en cuenta para producir el resumen final

durante la generación. [Over y otros, 2008]

La generación es el proceso para la creación de resumen de video [Valdés y Martínez, 2008, Over y otros, 2008, Sasonkgo, 2011]. Dicho proceso debe garantizar la clasificación, bajo determinados criterios, que permitan identificar la implicación de cada unidad básica de procesamiento en el resumen final. Esta clasificación permite determinar cuál o cuáles de las unidades básicas de procesamiento de la secuencia original deben incluirse en el resumen de video a generar. Un tercer paso [Valdés y Martínez, 2008] que se considera opcional es la presentación donde se realiza el procesamiento necesario para lograr el resumen final.

1.5.1. Modalidades de un resumen automático de video

En el estudio realizado por [Ngo y otros, 2003, Truong y Venkatesh, 2007] se establecen dos modalidades para la representación de los resúmenes de video que se pueden observar.

- Resúmenes construidos con imágenes estáticas, que constituyen una representación de los fotogramas relevantes extraídos a la secuencia de video original para mostrar el contenido de la misma a través de un guión gráfico estático, lo que comúnmente se conoce en la literatura anglosajona como *storyboard*. Esta modalidad es definida por [Truong y Venkatesh, 2007] de la siguiente forma: $R = A_{keyframe}(V) = F_i, F_{i+1}, ..., F_{i+n}$ donde $A_{keyframe}$ constituye el procedimiento de extracción de los fotogramas principales al video V, obteniéndose una representación R constituida por los fotogramas $F_i, F_{i+1}, ..., F_{i+n}$.

- Resúmenes que se componen de pequeños segmentos de video. Esta modalidad consiste en un conjunto de segmentos de video que se extraen del video original, los cuales se agrupan, ya sea por un corte o a través de un efecto gradual de transición, para obtener una secuencia de video con menor tamaño que el original. En este caso se constituye un pequeño video conocido en la literatura anglosajona como *skiming* de video. El resumen de video constituido por pequeños segmentos es definido por [Truong y Venkatesh, 2007] de la siguiente forma: $R = A_{skim}(V) = S_i \cup S_{i+1} ... \cup S_{i+n}$ donde A_{skim} constituye el procedimiento de generación del resumen del video V, S_i es el fragmento iésimo a incluir en el video resumen y $\cup$ la operación de integración, obteniéndose una representación R constituida por la integración de los segmentos $S_i, S_{i+1}, ..., S_{i+n}$.

1.5.2. Enfoques de las aproximaciones para generar resúmenes de video

Las investigaciones desarrolladas sobre resumen de video poseen diversos enfoques. Según [Truong y Venkatesh, 2007, Valdés y Martínez, 2008] existe una convergencia entre las aproximaciones más extendidas que se pueden clasificar en dos grupos principales:

- Los métodos que utilizan contenidos específicos de videos para los que se emplean patrones de comportamientos característicos según la tipología del contenido analizado, por ejemplo noticias, deportes o espectáculos, [Emna Fendri, 2010]
- Los métodos que utilizan información específica del contenido del video y se basan en técnicas de agrupamiento. Las técnicas de agrupamiento se emplean para eliminar las redundancias del contenido en el video y agrupar las unidades básicas de procesamiento más cercanas, según cierta clasificación establecida o su relación visual. [Dumont y Mérialdo, 2008]

1.5.3. Resumen escalable

El término ha sido utilizado en diversos campos de investigación con definiciones distintas adaptándose a la interpretación de cada autor. La idea que prevalece es que "algo" es escalable cuando es capaz de adaptarse adecuadamente a diferentes condiciones. En el ámbito del procesamiento de video el término se ha ampliado con la aparición de estándares de codificación escalables que permiten varias versiones de la secuencia, y se utiliza la misma codificación aunque varíe el flujo de datos.

Los investigadores en [Herranz y Martínez, 2010, Herranz Arribas, 2010, Herranz y otros, 2012] utilizan las características de la codificación escalable de una secuencia de video para la creación de resúmenes de varias longitudes.

En [Herranz y Martínez, 2010] se define que un resumen escalable es aquel que se constituye mediante un grupo de sumarios embebidos, $S = S_l, S_{l+1}, ..., S_{L-1}, S_L$ donde $l \in N$ denota la escala del resumen y L la longitud del mismo. A su vez se establece necesario el cumplimiento de la siguiente restricción $S_l \subset S_{l+1} \subset S_{L-1} \subset S_L$, considerando que cada sumario de menor escala es embebido en el sumario de escala superior.

El autor de la presente investigación considera que la definición anterior satisface las características de un resumen escalable. Como se puede apreciar el sumario de mayor longitud se conforma mediante la agrupación de los sumarios de menor longitud, por lo que se debe lograr una jerarquía

en la que el sumario de menor longitud debe lograr la mayor representatividad de la secuencia original.

1.6. Estudio de soluciones existentes

1.6.1. Exploración de la estructura de contenidos de un video para generar un resumen jerárquico

La solución propuesta por Zhu y sus colaboradores, [Zhu y otros, 2003] satisface la etapa de análisis desarrollando una técnica de detección para el dominio de codificación del Grupo Experto de Imágenes en Movimiento, del inglés, *Moving Picture Experts Group* (MPEG). En esta investigación se construye un método adaptativo que ajusta la detección de tomas a un umbral definido según las diferentes actividades en la secuencia de video. Debido a la incapacidad de esta técnica para adaptar el umbral a tomas diferentes dentro de la misma secuencia, se asume la utilización de una pequeña ventana de dos o tres segundos aproximadamente para establecer un umbral adaptado a la representación visual local de la ventana, lo que permite la detección de los cambios de tomas.

Una vez segmentadas las tomas, se extraen los fotogramas claves y calcula secuencialmente la similitud entre los fotogramas consecutivos $F_i, F_{i+1}, F_{i+2}, ..., F_{i+n}$ de la toma K, siendo i la posición del fotograma en K. Si se obtiene un valor de similitud menor que el umbral definido para la toma a la que pertenecen los fotogramas, entonces se incorpora el fotograma al grupo de fotogramas claves de la secuencia.

La similitud visual entre dos fotogramas es hallada utilizando tres descriptores visuales. Se calcula un histograma de color en el espacio HSV, se computa un descriptor de textura y otro de textura direccional.

Una vez establecidos los fotogramas claves, se explora su correlación mediante una matrix proximidad. La matriz permite determinar la relación visual entre los fotogramas claves para luego fusionar aquellos visualmente similares en un grupo. También se considera la información de orden temporal entre las tomas y se asume que las tomas con características visuales similares tienen una alta probabilidad de pertenecer a una escena, las tomas con mayor distancia temporal poseen más probabilidad de pertenecer a escenarios distintos. Las tomas con una distancia temporal mayor a un umbral predefinido, son divididas en diferentes grupos. De esta forma, se

agrupan las tomas adyacentes temporalmente, creando unidades que se consideran con mayor información semántica. Esta estrategia permite crear grupos que poseen características visuales similares al mismo tiempo que son adyacentes temporalmente. Posteriormente se establecen las escenas del video fusionando grupos que se consideran vecinos por su similitud temporal y visual. Durante la generación, para seleccionar los grupos representativos, se asume que aquellos con menor cantidad de tomas deben contener menos información, por lo que los grupos con una sola toma son despreciados. Se confeccionan los resúmenes de más de una toma utilizando dos métodos, uno estático para los resúmenes de menor nivel y otro dinámico para los resúmenes de mayor nivel. En el caso de los resúmenes de menor nivel, se construyen tomando todos los fotogramas claves existentes en los niveles más bajos del agrupamiento conformado. Mientras que para el resumen de mayor nivel se seleccionan cierto número de fotogramas, teniendo en cuenta las especificaciones del usuario final en cuanto a la longitud.

1.6.2. *Framework* para la generación de resúmenes escalables de video

Por su parte en el método propuesto en [Herranz y Martínez, 2010, Herranz Arribas, 2010] se asume directamente la generación del resumen a partir de la extracción del flujo de datos en videos codificados bajo los estándares del MPEG. Esta particularidad restringe el dominio de codificación en el que se puede aplicar, aunque permite asumir conceptos relacionados con las características [Mitchell y otros, 1996] de esta compresión que favorecen la propuesta.

Como este método está restringido a los estándares del MPEG, durante el análisis se utiliza como unidad básica de procesamiento el Grupo de Imagénes, del inglés, *Group of Pictures* (GOP) y consecuentemente los fotogramas convencionales de estos estándares (I, B y P) [Mitchell y otros, 1996]. Se asume que la secuencia original está compuesta por M GOPs y cada uno de estos a su vez posee el fotograma (I), que se toma como fotograma principal.

Lo primero que se realiza es recorrer la secuencia para obtener los GOPs, descartando todos los que poseen transiciones, para lo cual se clasifican los GOPs obtenidos y se eliminan los que poseen cambios de tomas y corta duración, considerando que ambos casos no contienen la información necesaria para su posterior inclusión en un resumen.

Cada toma es representada de forma uniforme, considerando la cantidad máxima de fotogramas claves que pertenecen a la misma. Se determina el descriptor *color layout* [Jalab, 2011] correspondiente a los fotogramas principales confeccionando el vector de características respectivo.

Una vez que se ha segmentado la secuencia de video original en GOPs, con los fotogramas claves correspondientes y se poseen los valores del descriptor *color layout* [Jalab, 2011], se eliminan las redundancias entre los fotogramas claves mediante la agrupación de aquellos con características visuales similares. Se obtiene una agrupación de K grupos $C_0, C_1, ..., C_{k-1}$, según las distancias entre cada par de fotogramas claves $d(K_i, K_j)$, que se hallan utilizando el descriptor *color layout*. Luego para cada grupo C_k el fotograma representativo es el más cercano al centroide.

Con el objetivo de obtener una representación escalable, se ordenan los grupos de manera incremental según la relevancia, obteniéndose dos listas jerárquicas: una para la modalidad de resúmenes estáticos y la otra para los dinámicos. En este caso la propiedad de resúmenes embebidos propuesta por los autores, descrita en el epígrafe 1.5.3, constituye la base de la jerarquía que establecen.

La relevancia de los grupos se define sobre la base de dos criterios, la duración y la distancia, combinados en una suma ponderada. La duración favorece a la selección de los grupos con mayor contribución en términos de duración de la secuencia, suponiendo que las agrupaciones más largas deben incluirse en las escalas inferiores. La distancia entre cada grupo se calcula teniendo en cuenta la similaridad de sus fotogramas claves representativos y favorece a la selección de los grupos con características visuales diferentes.

La clasificación se realiza de forma iterativa para cada GOP mediante dos posibles acciones: incluyendo un GOP adicional a una nueva toma o aumentando el GOP previamente incluido. Luego realizan la selección de los fotogramas relevantes y los GOPs. Cuando la longitud del resumen es muy limitada se logra la cobertura semántica básica de la secuencia utilizando la menor representación posible. Se supone que los grupos construidos contienen una relación semántica razonablemente buena y que cada grupo puede ser representado por un fotograma clave, para los resúmenes estáticos, y por una breve representación de GOPs consecutivos para los dinámicos. Se toma un fotograma principal o un segmento por cada grupo, para la generación de un resumen adecuado en condiciones de longitud muy limitada.

Aunque asumir los GOPs como unidad básica de procesamiento no es novedoso en investigaciones de procesamiento de video. Para la generación de resúmenes videos, esto constituye un elemento distintivo en relación a los métodos presentados en [Truong y Venkatesh, 2007], pues normalmente se asume el fotograma clave como unidad básica de procesamiento.

1.6.3. Valoración de las aproximaciones analizadas

En la aproximación [Zhu y otros, 2003] los autores obtienen buenos resultados aunque se asume la escalabilidad para cada estructura jerárquica con la limitante de que los resúmenes no son escalables dentro de cada nivel. A consideración del autor un resumen escalable debería lograr un mayor número de escalas para adaptarse a situaciones en las que, por ejemplo, la duración del mismo puede limitarse a un valor específico o porcentual. En el caso de la aproximación propuesta en [Herranz y Martínez, 2010, Herranz Arribas, 2010], los propios autores manifiestan que la incorporación de descriptores de video de bajo nivel para el análisis de la secuencia, debe mejorar sus resultados. La principal dificultad observada en la aproximación estudiada en el epígrafe 1.6.2, es que se ajusta el método a un dominio restringido de codificación de video que se basa en estándares del MPEG, y se centra en las características propias del flujo de datos para el análisis. Se considera que este elemento limita su utilización en otros dominios de codificación. Los aportes expuestos en [Herranz y Martínez, 2010, Herranz Arribas, 2010] constituyen, según las valoraciones del autor, los más acertados para la creación de resúmenes escalables de video.

1.7. Conclusiones parciales

- Las investigaciones precedentes explotan los descriptores de bajo nivel y generalmente su procesamiento se enmarca en una estrategia basada en fotogramas claves.
- Las modalidades de representación de un resumen de video que comúmente se observan en la literatura son: el resumen estático y el dinámico.
- En las aproximaciones existentes para generar resúmenes automáticos de video prevalecen los enfoques que basan su procesamiento en el contenido del video y la utilización de agrupamiento jerárquico.
- Las investigaciones que preceden este trabajo han establecido que para obtener resúmenes de video se realizan dos etapas fundamentales, análisis y generación.
- La aproximación propuesta en [Zhu y otros, 2004] y [Herranz y Martínez, 2010, Herranz Arribas, 2010], no se consideran pertinentes para solucionar la problemática que originó la investigación porque no se ajustan a diversas codificaciones.

Capítulo 2

Solución propuesta

En el presente capítulo se describe detalladamente el método propuesto para dar solución al problema de investigación planteado. Inicialmente se ofrece una visión general del método diseñado para obtener resúmenes escalables. Posteriormente se describe cómo realizar las etapas establecidas en el método y sus fases respectivas.

2.1. Método para la generación de resúmenes escalables

Para la concepción del método propuesto en el presente trabajo, se asumen los principios de las siguientes investigaciones [Valdés y Martínez, 2008, Herranz y Martínez, 2010, Herranz Arribas, 2010]. Se establecen dos etapas: análisis y generación. Se divide el procesamiento en dos etapas porque se debe obtener toda la información necesaria en la etapa de análisis, es decir, la representación del contenido del video. Lo anterior permitirá garantizar la condición de escalabilidad de los resúmenes. [Herranz y Martínez, 2010, Herranz Arribas, 2010] Una vez que se realice la etapa de análisis de una secuencia de video, cuando se requiera otro resumen de la misma secuencia, solamente será necesario realizar la etapa de generación. De esta forma se garantiza que cuando se ha analizado la secuencia de video original, es posible generar varios resúmenes de la misma, dependiendo de las condiciones de longitud establecidas. Para cada etapa se establecen las fases necesarias. Durante la etapa de análisis se realizará una fase de pre-procesamiento, la de segmentación y posteriormente la de agrupamiento. Con los resultados del análisis se debe realizar la etapa de generación, que a su vez cuenta con las fases de selección y creación.

2.2. Etapa de análisis

En la etapa de análisis se realiza el pre-procesamiento de la secuencia original. Se ha definido el pre-procesamiento como una fase, ya que se realizan todas las operaciones necesarias para adecuar la secuencia de video original a las condiciones requeridas para su posterior procesamiento.

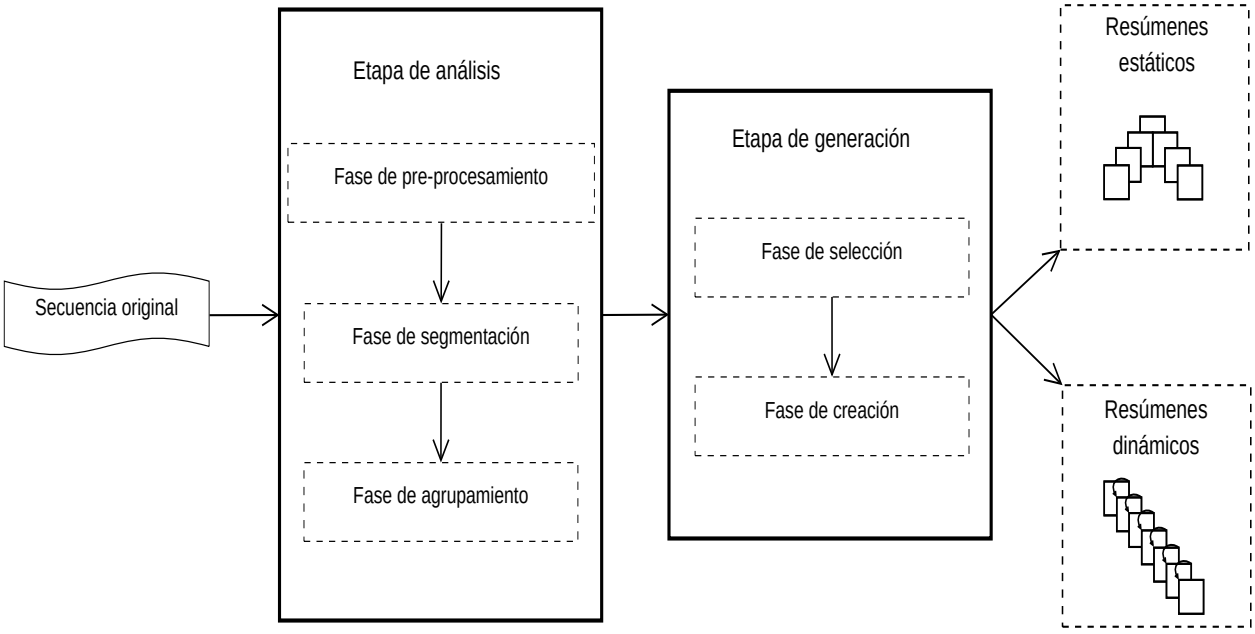

Figura 2.1: Representación del método para la generación de resúmenes escalables de video.

2.2.1. Fase de pre-procesamiento

Durante la fase de pre-procesamiento se extraen todos los fotogramas de la secuencia de video. La definición de un fotograma se ofreció en el capítulo 1, para la representación de su información visual se utiliza la definición 1, por lo que se obtiene la formulación:

$$F_i = c_1, c_2, ..., c_n \tag{2.1}$$

Donde F_i representa el fotograma en la posición i de una secuencia de video; su información audiovisual está caracterizada por el conjunto de descriptores $c_1, c_2, ..., c_n$ y n representa la cantidad de características del fotograma que se tendrán en cuenta. Para el método propuesto $n = 2$, pues se tendrán en cuenta dos descriptores: histograma de color y bordes.

Se recorre la secuencia de video para extraer todos los fotogramas que conforman la misma y los descriptores histogramas de color y bordes para cada fotograma. Ver pseudocódigo en Algoritmo 1.

Algoritmo 1 Extraer los fotogramas de una secuencia de video

```
1: function ExtractFrames(secuence type: VideoSecuence)
2:     mapframes type: Map < Integer, CFrame >
3:     frame type: CFrame
4:     framedata type: Mat
5:     reducedframe type: Mat
6:     vectorhsv type: Mat
7:     vectoredges type: Mat
8:     capture type: VideoCapture
9:     n ← frames ∈ secuence
10:    for i ← 0, i < n, i++ do
11:        if !Capture.read(secuence) then
12:            framedata ← capture.Read(s)
13:            reducedframe ← Resize(framedata)
14:            vectorhsv ← ComputeHitograms(reducedframe)
15:            vectoredges ← ComputeEdges(framedata)
16:            frame.CFrame(i, vectorhsv, vectoredges)
17:            mapframes.Insert(i, frame)
18:            Write(framedata, mapframes.Value(i))
19:        end if
20:    end for
21: end function
```

Extracción de histogramas de color

Como se menciona en el capítulo 2, en las aproximaciones propuestas por [Don y Uma, 2009, Qifan y otros, 2013, Zhang y Wang, 2012, Ejaz y otros, 2012, Mohanty y Kanungo, 2013] se considera que los histogramas de color permiten representar el contenido visual de los fotogramas. Además es una de las formas de representación audiovisual que menor complejidad computacional implica.

En la solución, se selecciona el espacio de color HSV [Ejaz y otros, 2012, Qifan y otros, 2013]. Una de las ventajas al utilizar este modelo, es que la componente de intensidad se separa de la información de color y las otras dos componentes proveen una representación del color que constituye la que más se asemeja al sistema humano de percepción del color, propiedades que convierten este espacio de color *"en muy apropiado para el procesamiento de imágenes basados en las características del sistema de visión humana"* [Soriano, 1998]. Además, un estudio experimental anterior [Pickering y Rüger, 2003], refiere que existen diferencias significativas al seleccionar un espacio de color y que HSV es el mejor a utilizar, por sus condiciones de ser perceptualmente

uniforme [González y Woods, 2002, Acharya y Ray, 2005].

Los fotogramas de la secuencia de video están en el espacio de color RGB, para su conversión al HSV, se normalizan las componentes (R, G y B) obteniéndose los valores (r, g, b) en un rango entre cero y uno. Luego se hallan los valores de las componentes H, S y V aplicando la formulaciones propuestas por [Acharya y Ray, 2005]:

$$V = max(r, g, b) \tag{2.2}$$

$$S = \begin{cases} 0 & \text{si} \quad V = 0 \\ V - \dfrac{min(r, g, b)}{V} & \text{si} \quad V > 0 \end{cases} \tag{2.3}$$

$$H = \begin{cases} 0 & \text{si} \quad S = 0 \\ \dfrac{60 * (g - b)}{S * V} & \text{si} \quad V = r \\ 60 * \left[2 + \dfrac{(b - r)}{S * V}\right] & \text{si} \quad V = g \\ 60 * \left[4 + \dfrac{(r - g)}{S * V}\right] & \text{si} \quad V = b \end{cases} \tag{2.4}$$

$$H = H + 360 \quad \text{si} \quad H < 0 \tag{2.5}$$

Para la obtención de un histograma de color más compacto se utiliza un histograma por cada canal H, S y V, lo que permite reducir la complejidad computacional en el cálculo posterior necesario para determinar las similitudes entre fotogramas. Se utiliza un histograma en una dimensión y se establece una longitud igual a 60 *bins* para la componente H. Asimismo para las componentes S y V se utiliza un histograma de 50 *bins*. La componente H puede alcanzar valores entre $[0, 360]$. Por otra parte los valores de las componentes S y V se hallan según una escala que se debe definir entre $[min_s, max_s]$ y $[min_v, max_v]$ respectivamente. Estos valores normalmente se establecen en los rangos entre $[0, 1]$, $[0, 100]$ ó $[0, 255]$ en dependencia de la implementación. Para la obtención de los histogramas se propone pseudocódigo en Algoritmo 2.

$$H = h_1, h_2, h_3, ...h_n | n = 60; \quad h \in \mathbb{N} \tag{2.6}$$

$$S = s_1, s_2, s_3, ...s_n | n = 50; \quad s \in \mathbb{N} \tag{2.7}$$

$$V = v_1, v_2, v_3, ...v_n | n = 50; \quad v \in \mathbb{N} \tag{2.8}$$

Algoritmo 2 Calcular el histograma de color para las componentes HSV

```
1: function ComputeHistograms(framedata type: Mat)
2:     vectorhsv type: Mat
3:     histh type: Mat
4:     hists type: Mat
5:     histv type: Mat
6:     histhsize ← 60
7:     histssize ← 50
8:     histvsize ← 50
9:     hrange ← (0, 360)
10:    srange ← (min_s, max_s)
11:    vrange ← (min_s, max_s)
12:    vectorchannels type: Mat
13:    SplitChannels(framedata, vectorchannels)
14:    Normalize(vectorchannels)
15:    ConvertToHSV(vectorchannels)
16:    ComputeHistogramH(vectorchannels.at(0), hish, histhSize, hrange)
17:    vectorhsv.PushBack(hish)
18:    ComputeHistogramS(vectorchannels.at(1), hists, histsSize, srange)
19:    vectorhsv.PushBack(hiss)
20:    ComputeHistogramV(vectorchannels.at(2), histv, histvSize, vrange)
21:    vectorhsv.PushBack(hisv)
22:    Normalize(vectorhsv)
23: end function
```

Determinación de formas, detector de contornos

Una de las principales dificultades que se observan en las aproximaciones que utilizan los histogramas de color como método de representación del contenido visual de la imagen, es que el color puede afectarse por cambios de iluminación. Por este motivo se decide extraer otro descriptor, en este caso basado en delimitar la forma de los objetos que conforman cada fotograma. [Lienhart, 1998, Lienhart, 2001]

Con el objetivo de localizar los bordes, que deben corresponderse con límites de los objetos

existentes, en los fotogramas se utiliza el detector de bordes propuesto por [Canny, 1986]. Este descriptor es utilizado en trabajos anteriores [Chandrakar y Bhonsle, 2012, McIlhagga, 2011, Ali y Clausi, 2001, Panetta y otros, 2011, Davies, 2012, McIlhagga, 2011, Panetta y otros, 2011] que refieren su efectividad para determinar las discontinuidades en la intensidad local de los píxeles.

Para aplicar el detector de Canny, [Canny, 1986, Boyle y Thomas, 1988, Bradski y Kaehler, 2008, Laganière, 2011, Davies, 2012] se convierte la imagen a escala de grises, para lo cual se realiza una transformación al espacio de color YIQ y se toma el valor de la componente Y como imagen en escala de grises. Este procesamiento se realiza por la formulación propuesta en [Acharya y Ray, 2005]:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0,299 & 0,587 & 0,114 \\ 0,596 & -0,274 & -0,322 \\ 0,211 & -0,523 & 0,312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{2.9}$$

$$Y = 0{,}299 * R + 0,587 * G + 0,114 * B \tag{2.10}$$

Posteriormente se calculan los gradientes de intensidad horizontal y vertical de la imagen. Se halla la magnitud y dirección de los gradientes y se eliminan los píxeles que pertenecen a bordes que no poseen un gradiente de magnitud máxima. Finalmente se determina la inclusión de los píxeles como parte de los bordes utilizando los umbrales superior e inferior:

- Si el gradiente de un píxel es mayor que el umbral superior, este píxel se acepta como parte del borde.
- Si el gradiente de un píxel es menor que el umbral inferior, entonces no se considera como parte del borde.
- Si el gradiente de un píxel se encuentra entre el umbral inferior y el superior, es aceptado si está conectado con un píxel que posee un gradiente por encima del umbral superior.

Como resultado de este procedimiento se obtiene un conjunto de píxeles con valor cero o uno, representando el color negro o blanco, los píxeles blancos determinan los bordes.

Para el caso de la caracterización basada en bordes, el descriptor se determina de forma local. Toda la imagen se divide en nueve bloques y se extraen los bordes por cada uno de estos bloques.

Las nueve regiones no se distribuyen de forma uniforme debido a que se le proporciona mayor importancia al área que se encuentra en la región central del fotograma. La división se realiza para poder analizar posteriormente los bordes presentes en cada una de estas regiones, y se explicará en el epígrafe 2.2.2. Para determinar los bordes de los fotogramas se propone el pseudocódigo en Algoritmo 3.

Algoritmo 3 Computar bordes

```
1:  function ComputeEdges(framedata type: Mat)
2:      vectoredges type: Mat
3:      vectorroi type: Mat
4:      vectorroi ←ComputeROI(framedata)
5:      length ← vectorroi.Size()
6:      for i ← 0, i < length, i++ do
7:          edges type: Mat
8:          edges ← ConvertToGrayScale(vectorroi.At(i))
9:          edges ← AplyGaussianBlur(vectorroi.At(i))
10:         edges ← Canny(vectorroi.At(i), lowthreshold, highthreshold)
11:         vectoredges.Pushback(edges)
12:     end for
13: end function
```

Una vez que se han obtenido al menos dos fotogramas del video, estos se deben procesar para definir los límites entre tomas. Se utiliza la información de los descriptores que caracterizan a los mismos. De esta forma comienza la fase de segmentación.

2.2.2. Fase de segmentación

En las aproximaciones propuestas para la detección de tomas se evidencia un paso inicial común. Consiste en la comparación de los fotogramas adyacentes en la secuencia de video mediante métodos definidos para determinar la semejanza entre dichos fotogramas [Bescós y otros, 2005, Sáez Peña, 2006, Lu y Shi, 2013, Chavan y otros, 2013].

Como se ha mencionado anteriormente, para la representación visual del contenido de cada fotograma se observan aproximaciones que utilizan métodos basados en histogramas de color y métodos basados en bordes [Smeaton y otros, 2010, Weiming y otros, 2011, Thounaojam y otros, 2014].

El objetivo que se persigue al ejecutar la etapa la segmentación es, a partir de los vectores que caracterizan el contenido visual de cada fotograma del video, determinar los límites entre tomas

de éste y establecer el fotograma principal para cada toma. Asumiendo 2.1, es posible realizar representación matemática de una toma como:

$$T = F_i, F_{i+1}, F_{i+2}, ..., F_{i+n} \tag{2.11}$$

Donde T representa la toma que está conformada por un grupo de fotogramas F, siendo n el número de fotogramas pertenecientes a la toma, así como (i) la posición de los mismos.
En el conjunto de fotogramas que conforman la toma, se establece el fotograma principal de dicha toma. Este debe poseer las características más representativas de todos los que pertenecen a la toma, lo que se explica con mayor detalle en un acápite dedicado a este proceso, ver 2.2.2.
Basado en que varias propuestas utilizan solamente una característica de la imagen para la segmentación, pero es posible observar otras que combinan varios desriptores [Chavan y otros, 2013], en la solución como ya se ha mencionado, se combinan los vectores hallados previamente, pues se ha comprobado [Sáez Peña, 2006] que mejora considerablemente la eficacia de los resultados. Se utilizan los conceptos de toma probable y toma real [Sáez Peña, 2006]. Se procesan secuencias utilizando los dos descriptores. Con uno se detectan los límites entre tomas mediante los histogramas de color. Se pretende obtener todos los límites entre tomas abruptas, pues como se explicó en el capítulo 2, una toma posee un fondo característico, por lo tanto, cuando hay cambios la información del color debe variar. Debido a que puede existir cierta sensibilidad a los cambios de iluminación, con el otro descriptor se persigue filtrar los límites detectados con histogramas de color y obtener los cambios entre tomas graduales que se introducen por efectos de edición, teniendo en cuenta los bordes.

Determinación de la similaridad entre fotogramas

Similaridad basada en histogramas de color

En el estudio de [Smeaton y otros, 2010], al referirse a la utilización de los histogramas de color para la segmentación, argumenta que la selección de las medidas de similaridad utilizada no posee gran influencia para determinar cuánto se parece un fotograma a otro. Igualmente se refiere por [Rubner y otros, 2001, Smeaton y otros, 2010], que en el dominio de procesamiento audiovisual las distancias de Manhattan y Euclidiana poseen efectividad, por lo que se utiliza la distancia Euclidiana en la solución 2.12 entre ambos histogramas de las componentes H, S, V de dos

fotogramas adyacentes $F_{t,i}, F_{t+1,i+1}$. [Bradski y Kaehler, 2008, Ejaz y otros, 2012]

$$f_d(F_{i+1}, F_i) = \sqrt{\sum_{0}^{60}(h_{n,i+1} - h_{n,i})^2 + \sum_{0}^{50}(s_{n,i+1} - s_{n,i})^2 + \sum_{0}^{50}(v_{n,i+1} - v_{n,i})^2} \tag{2.12}$$

Una vez que halladas las distancias para cada par de fotogramas $f_d(F_{i+1}, F_i)$ respectivamente, se considera que existe un cambio de toma probable cuando se cumple que:

$$f_d(F_{i+1}, F_i) \geq \gamma \tag{2.13}$$

En la solución se ha asumido $\gamma = 0,4$. Este valor de umbral se ha determinado realizando una validación cruzada con K cortes[1]. En este caso se utiliza específicamente la modalidad de validación cruzada dejando un elemento fuera[2] para obtener un modelo donde γ garantice disminuir el error estimado.[Hastie y otros, 2009, Witten y otros, 2011]

Similaridad basada en bordes

Para la estimación de la similaridad entre los bordes se trabaja con los bordes que caracterizan a los fotogramas y se determina la similaridad para establecer los límites entre tomas graduales. Además, se realiza un filtrado de las tomas probables que se detectaron con histogramas, debido a que este procedimiento puede introducir cambios inexistentes por variaciones de iluminación. La determinación de la similaridad basado en bordes se realiza mediante una ventana deslizante, estableciéndose una ventana deslizante W que estará constituida por un grupo de l fotogramas.

$$W = F_i, F_{i+1}, F_{i+2}, ..., F_{i+(l-1)} \tag{2.14}$$

En este caso como la descripciones que se tienen son locales es necesario realizar las comparaciones entre los bordes de cada bloque del fotograma para obtener la similaridad global.
Debido a que el descriptor aporta una imagen binaria que posee un conjunto de píxeles en blanco o negro, se calcula su similaridad basado en la matriz binaria que aporta esta imagen, hallando la similitud Coseno sobre vectores binarios, lo que se conoce también en la literatura como similaridad de Ochiai [Dunn y Everitt, 2004]. Para ello se tiene en cuenta la taxonomía de

[1] *K-Fold Cross-Validation*
[2] *Leave-One-Out Cross-Validation*

Algoritmo 4 Calcular Distancia Euclidiana entre dos fotogramas

```
1: function ComputeEuclidianDistance(frame_i type: CFrame, frame_{i+1} type: CFrame)
2:     distance ← 0 type: float
3:     hvalues ← 0 type: float
4:     svalues ← 0 type: float
5:     vvalues ← 0 type: float
6:     dims ← frame_i.GetVectorHSV().At(0).GetSize() type: Integer
7:     for i ← 0, i < dims, i++ do
8:         varhf_i ← frame_i.GetVectorHSV().At(0).At(i)
9:         varhf_{i+1} ← frame_{i+1}.GetVectorHSV().At(0).At(i)
10:        hvalues ← hvalues + (varhf_{i+1} − varhf_i)^2
11:    end for
12:    dims ← frame_i.GetVectorHSV().At(1).GetSize() type: Integer
13:    for i ← 0, i < dims, i++ do
14:        varsf_i ← frame_i.GetVectorHSV().at(1).At(i)
15:        varsf_{i+1} ← frame_{i+1}.GetVectorHSV().At(1).At(i)
16:        svalues ← svalues + (varsf_{i+1} − varsf_i)^2
17:    end for
18:    dims ← frame_i.GetVectorHSV().At(2).GetSize() type: Integer
19:    for i ← 0, i < dims, i++ do
20:        varvf_i ← frame_i.GetVectorHSV().At(2).At(i)
21:        varvf_{i+1} ← frame_{i+1}.GetVectorHSV().At(2).At(i)
22:        vvalues ← vvalues + (varvf_{i+1} − varvf_i)^2
23:    end for
24:    distance ←ComputeEuclidianDistance(hvalues, svalues, vvalues)
25:    return distance
26: end function
```

operaciones binarias propuesta en [Dunn y Everitt, 2004, Choi y otros, 2010].

Tabla 2.1: Taxonomía de operaciones binarias para instancias (i, j)

j \ i	1	0	Sum
1	$a = i \cap j$	$b = \bar{i} \cap j$	$a + b$
0	$c = j \cap \bar{i}$	$d = \bar{i} \cap \bar{j}$	$c + d$
Sum	$a + c$	$b + d$	$n = a + b + c + d$

En la taxonomía i y j se refieren a los valores que toman dos píxeles en la misma posición para dos fotogramas.

Finalmente para determinar la similaridad de Ochiai se utiliza la función de similitud propuesta

en [Choi y otros, 2010, Deza y Deza, 2012] que está dada por la siguiente formulación:

$$f_s(F_{i,j}, F_{k,j}) = \frac{\sum_1^n a}{\sqrt{(\sum_1^n a + \sum_1^n b)(\sum_1^n a + \sum_1^n c)}} \tag{2.15}$$

Donde n es la cantidad de píxeles de la región j para los fotogramas F_i, F_k. Debido a que la función de similitud es aplicada a cada región en la que se ha dividido el fotograma. Para obtener la similitud del fotograma completo se divide el valor de similitud hallado por cada bloque entre el total de bloques y se suman estos valores, siendo el resultado de esta sumatoria el valor de similaridad para los fotogramas.

$$f_s(F_i, F_k) = \sum_{j=1}^{9} f_s(F_{i,j}, F_{k,j})/9 \tag{2.16}$$

Para filtrar los límites probables detectados, se comprueba si el valor de similitud resultante, para un par de fotogramas está por debajo de γ y se ha detectado en esa misma posición un cambio por histograma probable. En caso positivo se considera un cambio de toma real. De lo contrario, si el valor no está por debajo de γ y existe un límite probable detectado en esa misma posición por histograma, éste se desecha. Para el filtrado se utiliza un $\gamma = 0{,}6$.

Si al comparar los fotogramas que están en los extremos de la ventana W, el resultado también se encuentra por debajo de γ y dentro de la ventana no se ha establecido un límite de toma probable, se establece en la posición intermedia de W un límite de toma. Para la detección de cambios graduales se utiliza un $\gamma = 0{,}4$.

Al igual que en el método de similaridad anterior, se ha asumido el valor γ después de realizar una validación cruzada con K cortes, utilizando la modalidad de validación cruzada dejando un elemento fuera.[Hastie y otros, 2009, Witten y otros, 2011]

Una vez detectadas todas las tomas estas serán filtradas. Si existen tomas, detectadas, que posean una longitud menor de 32 fotogramas o mayor que el 25 por ciento del total de fotogramas del video original, estas se eliminan porque se consideran tomas que no aportan información significativa sobre el contenido del video original.

Algoritmo 5 Calcular similitud de bordes entre dos fotogramas

```
1: function ComputeOchiaiSimilarity(frame_i type: CFrame, frame_l type: CFrame)
2:     similarity ← 0 type: float
3:     a ← 0 type: Integer
4:     b ← 0 type: Integer
5:     c ← 0 type: Integer
6:     cols ← 0 type: Integer
7:     rows ← 0 type: Integer
8:     length ← 9 type: Integer
9:     for i ← 0, i < length, i++ do
10:        cols ← frame_i.GetVectorEdges().At(i).Cols()
11:        rows ← frame_i.GetVectorEdges().At(i).Rows()
12:        for j ← 0, j < cols, j++ do
13:            for k ← 0, k < rows, k++ do
14:                var_i ← frame_i.GetVectorEdges().At(i).At(j, k) type: Integer
15:                var_l ← frame_l.GetVectorEdges().At(i).At(j, k) type: Integer
16:                if var_i = 1 & var_l = 1 then
17:                    a++
18:                end if
19:                if var_i = 0 & var_l = 1 then
20:                    b++
21:                end if
22:                if var_i = 1 & var_l = 0 then
23:                    c++
24:                end if
25:            end for
26:        end for
27:        similary ← similary+ComputeSimilarityOfRegion(a, b, c)
28:    end for
29:    return similary
30: end function
```

Determinación del fotograma principal

Como último paso en la fase de segmentación, se determina qué fotograma, en el grupo que conforma la toma es el más representativo, denominándolo como fotograma principal de la toma. Para la determinación del fotograma principal en la solución se toman los descriptores de contornos de las nueve regiones que se hallaron previamente. Estas regiones poseen esta distribución porque se asume que en las áreas (B, D, F, H) existe mayor actividad y por lo tanto aportan más información. Por cada uno de estos bloques se asigna una ponderación en una escala $1 \leq \rho \leq 5$ según la representación matricial que se especifica en 2.17. Se asignan los valores superiores a las

regiones que se consideran de mayor interés. [Chan y otros, 2011]

$$\begin{bmatrix} A & B & C \\ D & E & F \\ G & H & I \end{bmatrix} = \begin{bmatrix} \rho = 1 & \rho = 2 & \rho = 1 \\ \rho = 3 & \rho = 5 & \rho = 3 \\ \rho = 2 & \rho = 4 & \rho = 2 \end{bmatrix} \tag{2.17}$$

Luego, para la determinación del fotograma principal, se halla la sumatoria de los bordes detectados en cada una de las regiones y se multiplica por el peso asignado a esta región. Los mayores valores de ρ se asignan a las regiones donde se considera debe existir la mayor actividad. Se realiza una comparación basada en la similitud de todos los fotogramas pertenecientes a la toma. Se seleccionan aquellos fotogramas que posean mayor similitud, dentro de estos el que aporte mayor cantidad de píxeles en blanco se selecciona como fotograma principal. En caso de que existan fotogramas similares y con la misma cantidad de píxeles en blanco se toma uno de estos de forma aleatoria. El pseudocódigo propuesto para calcular la similitud entre los fotogramas de la toma es muy similar al Algoritmo 5, pero en este caso se agrega el ρ a la fórmula de similaridad y el cálculo de la cantidad de píxeles en blanco.

$$f_s(F_i, F_k) = \sum_{j=1}^{9} \rho * f_s(F_{i,j}, F_{k,j})/9 \tag{2.18}$$

2.2.3. Fase de agrupamiento

Se realiza un agrupamiento jerárquico utilizando una técnica de selección aglomerativa y la estrategia utilizada es el enlace completo[3]. [Xu y C. Wunsh, 2009]

La matriz de similitudes entre los fotogramas se forma con $N \times N$ dimensiones donde N será igual al número de tomas, pues se corresponde con la cantidad de fotogramas principales. Para lograr los valores de similitud en cada posición de la matriz se representa el valor alcanzado al hallar la distancia por una formulación que combina, la distancia Euclidiana 2.12 entre los histogramas de color y la distancia de Ochiai 2.15, para los descriptores de bordes en los fotogramas principales. Como toda matriz de similitud, ver 2.21, basada en una función de distancia, en la diagonal se tienen siempre el valor 0 que representa la máxima similaridad.

[3] *Complete Linkage (farthest neighbor)*

Debido a que en las funciones de similitud, el valor que expresa la mayor semejanza es $f_s \approx 1$ y en las funciones de distancia, la mayor semejanza es $f_d \approx 0$, como se explicó en el capítulo anterior, se utiliza la distancia complemento de la función de similitud $1 - f_s$. Luego para determinar los valores de similitud en la matriz se utiliza la formulación 2.20. Para esta formulación se han definido las variables μ_d y μ_s, como ponderaciones que se le asignan a las funciones utilizadas, según la importancia que se le confiere a cada métrica para establecer la similaridad. Debe cumplirse que:

$$0 \leq \mu_d + \mu_s \leq 1 \tag{2.19}$$

$$S = \mu_d * f_d(F_i, F_k) + \mu_s * (1 - f_s(F_i, F_k)) \tag{2.20}$$

$$\begin{bmatrix} & F_1 & F_2 & F_3 & ... & F_n \\ F_1 & 0 & ... & ... & ... & ... \\ F_2 & ... & 0 & ... & ... & ... \\ F_3 & ... & ... & 0 & ... & ... \\ ... & ... & ... & ... & 0 & ... \\ F_n & ... & ... & ... & ... & 0 \end{bmatrix} \tag{2.21}$$

Una vez obtenidos los valores de similitud entre todos los fotogramas, se prosigue con la construcción de la estructura que soporte el agrupamiento jerárquico y se crean los grupos. Cada grupo estará compuesto por el valor de similitud que los une y por dos elementos, que pueden ser fotogramas o subgrupos. Para obtener el primer grupo se computa la distancia entre cada coordenada (i, j) de la matriz y los fotogramas que posean menor distancia serán agrupados, conformando un grupo que posee dos fotogramas y el valor de similitud entre estos. De esta forma se agrupan tomando cada par de valores más cercanos a cero, lo que representa la mayor similaridad. Cuando se crea un grupo, éste se convierte en un nuevo elemento que toma el valor de similitud por el que se conformó. Se continúan agrupando de forma iterativa los fotogramas o los grupos conformados a un nivel que se corresponde con el valor de similitud por el que se agrupan. Mientras existan grupos o fotogramas que no estén unidos a ningún grupo se repite el proceso de forma combinatoria hasta lograr la estructura jerárquica.

Algoritmo 6 Agrupamiento jerárquico aglomerativo

```
1: function GroupFrames(mapframes type: Map < Integer, CFrame >)
2:     dims ← mapframes.Values().Size() type:Integer
3:     matrix[dims][dims] type:float
4:     for i ← 0, i < dims, i + + do
5:         matrix.At[i, i] ← 0
6:         for j ← 1, j < dims, j + + do
7:             matrix.At[i, j] ← ComputeSimilarity(mapframes.Values().At(i),mapframes.Values().At(j)))
8:         end for
9:     end for
10:    while matrix.Size()≠ 1 do
11:        temp ← 1 type:float
12:        k ← 0 type:Integer
13:        n ← 0 type:Integer
14:        for i ← 0, i < dims, i + + do
15:            for j ← 0, j < dims, j + + do
16:                if i ≠ j then
17:                    if matrix.At[i, j] < temp then
18:                        temp ← matrix.At[i, j]
19:                        k ← i
20:                        n ← j
21:                    end if
22:                end if
23:            end for
24:        end for
25:        Join(mapframes.Value(k),mapframes.Value(n)
26:        UpdateMatrix(matrix, k, n)
27:    end while
28: end function
```

2.3. Etapa de generación

En la etapa de generación se realizan todas las operaciones necesarias para lograr la representación final del resumen. Esta se compone por las fases de selección y creación del resumen respectivamente, que se explican en los siguientes epígrafes.

2.3.1. Fase de selección

En la selección se determinan las unidades que formarán parte del resumen, para lo que se explota la estructura jerárquica construida durante el análisis. Debido a que ésta se conformó utilizando una estrategia de unión simple y se pretende lograr la mayor cobertura semántica posible, se

seleccionan los fotogramas que poseen la menor similitud. Para ello se realiza un recorrido sobre la estructura en busca de los elementos que se encuentran ubicados a mayor y menor nivel.
Para lograr el resumen de menor escala se toman los dos grupos más alejados, que están en correspondencia con los fotogramas que se encuentran agrupados en niveles superiores y los que están agrupados en niveles inferiores. Para el nivel superior se busca el grupo que contenga al menos un fotograma, en caso de que el grupo contenga dos fotogramas se toma uno de estos de forma aleatoria. Para el nivel inferior, se busca aquel grupo que contenga dos fotogramas, luego se considera la inclusión de cualquiera de los dos de forma aleatoria. Una vez que se tienen los dos elementos más distantes, se puede asumir su representación como el resumen de menor longitud o escala respectivamente. Si se requiere aumentar la escala del resumen, se va buscando por los niveles de la estructura jerárquica los elementos que se encuentren a mayor distancia, cuando se alcanzan dos nuevos elementos a incluir, éstos se consideran en una escala mayor que a la vez constituye un subconjunto de la escala anterior.
El proceso anterior garantiza el resumen de menor escala pero se debe aumentar la escala de ser necesario. Para esto hay que tener en cuenta las preferencias de longitud, pues como se ha explicado anteriormente, es el criterio de escalabilidad que se sigue en la investigación, ver 1.5.3. Para determinar las preferencias de longitud se requiere el por ciento del video original que se desea en el resumen resultante. Una vez establecido el valor porcentual que se requiere, se calcula la cantidad de unidades básicas necesarias para satisfacerlo. Luego se aplica la estrategia de búsqueda sobre el agrupamiento jerárquico. En este caso el criterio de parada está determinado por el valor de unidades a representar, es decir, cuando se obtiene la cantidad de elementos representativos necesarios se termina la búsqueda y estos quedan establecidos como los fotogramas representativos para satisfacer la especificación de longitud.
El resultado final de la selección es una lista ordenada con los fotogramas que deben formar parte del resumen. La lista se ordena teniendo en cuenta la similitud entre los fotogramas principales que la conforman. Estos se ubican ordenados consecutivamente de menor a mayor por los valores de similitud entre cada uno, con el objetivo de lograr la menor dispersión semántica posible.

2.3.2. Fase de creación

Durante la etapa de creación se confecciona finalmente el resumen resultante. Como resultado de la selección se poseen los fotogramas representativos que formarán parte del resumen.

Algoritmo 7 Selección

```
1: function GetFrames(size type:Integer)
2:     list type:List < Integer >
3:     if size ≤GetNumberOfSubnodes() then
4:         Find(list, size)
5:     else
6:         Find(list,GetNumberOfSubnodes())
7:         list.Add(this.GetIndex())
8:     end if
9:     return list
10: end function
11: function Find(size type:Integer,list type:List < Integer >)
12:     if isLeaf() then
13:         list.Add(this.GetIndex())
14:     else
15:         var_a ←< Integer > size/2 type:Integer
16:         var_b ← a + size %2 type:Integer
17:         if var_a >Left().GetNumberOfSubnodes() then
18:             var_b ← var_b+(Left().GetNumberOfSubnodes()−var_a)
19:             var_a ←Left().GetNumberOfSubnodes()
20:         else if var_b >Right().GetNumberOfSubnodes() then
21:             var_a ← var_a+(Left().GetNumberOfSubnodes()−var_b)
22:             var_b ←Right().GetNumberOfSubnodes()
23:         end if
24:         if var_a > 0 then
25:             Left().Find(var_a, list)
26:         end if
27:         if var_b > 0 then
28:             Right().Find(var_a, list)
29:         end if
30:     end if
31: end function
```

Creación de resúmenes estáticos

La modalidad más sencilla es la estática. Para confeccionar un resumen estático, simplemente se representan los fotogramas que se encuentran en la lista ordenada, conformando una secuencia de imágenes estáticas. Los datos que conforman la información visual del fotograma se encuentran almacenados en el directorio que se determinó para su almacenamiento, por lo que se debe acceder al mismo y buscar para este fotograma su información visual correspondiente. Debido a que cada fotograma, durante su extracción, fue identificado como un fichero con su índice en el video original y se ha trabajado con este identificador durante todo el proceso; solamente es

necesario realizar un búsqueda sobre el directorio donde el nombre de los archivos coincida con el índice de los fotogramas representativos.

Con el objetivo de reducir el tiempo de cómputo necesario para esta búsqueda se aprovechan las arquitecturas de los procesadores modernos explotando los recursos computacionales simultáneamente. [Gove, 2010]

Algoritmo 8 Búsqueda

```
1:  function FindFrames(listframesindex type: List < Integer >, path type: String)
2:      index ← " "type:String
3:      dir ← path type:Dir
4:      dir ← dir.Dir(path)
5:      files ← dir.Files() type:List
6:      parallel for private (j) shared (flag)
7:      for i ← 0, i < listframesindex.Length(), i + + do
8:          flag ← false type: Boolean
9:          index ←< String > listframesindex.At(i)
10:         for j ← 0, j < files.Length(), j + + do
11:             if flag ≠ true then
12:                 continue
13:                 if ficheros.At(j).Equals(index+".jpeg")) then
14:                     Present(path + index+".jpeg")
15:                     flag ← true
16:                 end if
17:             end if
18:         end for
19:     end for
20: end function
```

Creación de resúmenes dinámicos

Para el caso de los resúmenes dinámicos, sí es necesario realizar otro grupo de operaciones. Como entrada se tiene igualmente la lista de fotogramas representativos a mostrar en el resumen. Es necesario recuperar las tomas a las cuales representan estos fotogramas principales. Una vez que se poseen las tomas, cada una contiene una colección con los índices de los fotogramas que la conforman. Para recuperar todos los fotogramas de la toma se realiza la misma búsqueda que se explicó en el epígrafe anterior, en este caso con mayor coste computacional debido a que se necesita recuperar mayor cantidad de fotogramas. Se confecciona una lista con los fotogramas organizados por tomas y éstas a su vez se mantienen organizadas en correspondencia con la

definición realizada durante la selección al crear la lista de fotogramas principales. Finalmente, se recorre la lista de fotogramas uniéndolos para confeccionar el video correspondiente a la modalidad de resumen dinámico.

2.4. Conclusiones parciales

- La división del método en etapas y fases posibilita analizar un video para obtener la información que representa el contenido del mismo y garantiza la generación de resúmenes de diversa longitud, sin necesidad de analizar nuevamente el video.
- La extracción y representación de la información audiovisual mediante los descriptores de bajo nivel de color y bordes, permitió determinar las diferencias entre fotogramas utilizando las funciones de distancia Euclidiana 2.12 y de Ochiai 2.15, así como determinar la matriz de similitudes para el agrupamiento jerárquico, combinando estas funciones mediante la distancia complemento de 2.15.
- La etapa de generación establecida garantiza la obtención de las dos modalidades de representación de un resumen que se observan comúnmente en la literatura, los estáticos y los dinámicos.

Capítulo 3

Validación de la solución

En el presente capítulo se describe el proceso de validación del método propuesto para dar solución al problema de investigación planteado. Se reseña un componente desarrollado tomando el método como base y se analizan los resultados con el objetivo de validar primeramente la fase de segmentación de la etapa de análisis. Posteriormente se aplica un instrumento de diagnóstico para comprobar la escalabilidad y usabilidad de los resúmenes resultantes. Además se realiza una comprobación del método para distintas codificaciones y se evalúa su rendimiento en comparación con otra aproximación.

3.1. Desarrollo del componente para generar resúmenes escalables de una secuencia de video

Se ha desarrollado un componente informático, que tomando como base el método propuesto en el capítulo anterior, permite procesar una secuencia de video transitando por las etapas y fases establecidas. El desarrollo del componente se realizó utilizando el lenguaje de programación C++ [1995, Alexandrescu, 2001], sobre el Entorno de Desarrollo Integrado, del inglés, *Integrated Development Environment* (IDE), *Qt Creator*, utilizando las potencialidades que brinda el *Framework* Qt en su versión 4.8[1], integrando igualmente las funcionalidades de la biblioteca de procesamiento de imágenes y señales OpenCV [Bradski y Kaehler, 2008, Laganière, 2011, Laganière, 2014] en su versión 2.3.1; además se utiliza en determinados procedimientos la Interfaz de Programación de Aplicación, del inglés, *Application Programming Interface* (API), OpenMP, estandarizada para la programación en paralelo sobre sistemas de memoria compartida [Acevedo Martínez, 2013]. Para la implementación del componente y las pruebas realizadas que se exponen en otras secciones del capítulo, se utilizó un ordenador personal que posee microprocesador Intel Core i3 M350 a 2.27 GHz, con una Memoria de Acceso Aleatorio, del inglés, *Random Access Memory* (RAM), de 4 GB y como sistema operativo la distribución de Linux: Ubuntu 12.04.

[1] http://qt-project.org/

3.2. Validación de la fase segmentación en la etapa de análisis

Diversos referentes teóricos que se pueden encontrar en la literatura [Su, 1994, Powers, 2011, Gómez Carranza y Moens, 2014], develan que en toda aplicación donde se pretenda el procesamiento de datos e información exento de intervención o supervisión humana, es necesario establecer claramente los índices de precisión del sistema en cuestión. Desde luego que las aplicaciones de procesamiento automático de video requieren precisión y en aproximaciones precedentes [Arbelaez y otros, 2009, Castellanos y otros, 2010, Brutzer y otros, 2011, Lee y otros, 2011b, Klicnar y Beran, 2012] se evidencia la medición de este indicador para evaluar los resultados de sistemas automatizados de procesamiento de video.
En el caso de la investigación se mide el indicador de precisión para la etapa de análisis durante la fase de segmentación, debido a que es donde se garantiza la obtención de todos los datos necesarios para la posterior generación del resumen. Se determinó validar esta etapa de forma independiente, porque la fase de segmentación constituye una sección crítica en el método propuesto.

3.2.1. Medición de *Precision-Recall* para la segmentación

En los problemas de clasificación binaria las clasificaciones pueden ser positivas o negativas. De esta forma la respuesta del clasificador se puede representar en una matriz de confusión. Para la confección de la matriz de confusión se tienen en cuenta cuatro valores: [Hernandez Heredia, 2013, Gómez Carranza y Moens, 2014]

- TP: total de fotogramas que son límites entre tomas y son detectados como tal .
- FN: se refiere a la cantidad fotogramas que son límites de tomas y no son detectados como tal.
- TN: cantidad de fotogramas que no son límites entre tomas y no se identifican como tal.
- FP: total de fotogramas que no son límites de tomas y son detectados como tal.

Pero para el caso de la segmentación el resultado que ofrece el TN no se considera significativo por lo que no se incluye. [Cózar, 2010]

Tabla 3.1: Matriz de confusión, fuente [Hernandez Heredia, 2013, Gómez Carranza y Moens, 2014].

	Detectado positivo	Detectado negativo
Real positivo	TP	FN
Real negativo	FP	-

Para computar las métricas de *precision* y *recall* se utilizan las formulaciones:

$$Recall = \frac{TP}{TP + FN} \tag{3.1}$$

$$Precision = \frac{TP}{TP + FP} \tag{3.2}$$

Como base de datos de prueba para la medición de *Precision-Recall* se toman cinco secuencias de videos documentadas en el reporte técnico realizado por [Bescós, 2003]. Las propiedades de los videos se pueden observar en la tabla 3.2.

Tabla 3.2: Propiedades de las secuencias seleccionadas para validar la segmentación, fuente [Bescós, 2003].

Archivo de video	Referencia	Duración (MM:SS)	Fotogramas por segundo (FPS)	Resolución (Ancho-XAlto)	Fotogramas	Cambios de tomas
News11	V-1	28:33	25 FPS	352X288	42828	286
News12	V-2	18:26	25 FPS	352X288	27400	103
Basket	V-3	15:38	25 FPS	352X288	23450	113
Cycling	V-4	19:13	25 FPS	352X288	28823	74
Drama	V-5	15:36	25 FPS	352X288	23390	148

Los resultados que recoge la tabla 3.3, a consideración del autor, evidencian la eficacia de la segmentación en la etapa de análisis que a su juicio es buena ya que se obtienen índices de *precision-recall* por encima de 0,86 en todos los casos.

Tabla 3.3: Resultados de las mediciones para *Precision-Recall*

Referencia	TP	FP	FN	*Precision*	*Recall*
V-1	249	27	37	0,90	0,87
V-2	89	11	14	0,89	0,86
V-3	97	12	16	0,89	0,86
V-4	67	8	7	0,89	0,91
V-5	128	18	20	0,88	0,86

3.3. Validación de la escalabilidad y la usabilidad de los resúmenes generados

Para garantizar la usabilidad de un resumen de video, se debe crear un sumario que posea un lenguaje audiovisual que cumpla con dos condiciones esenciales: cobertura semántica y agrado visual. La primera se refiere a que el resumen creado debe preservar la mayor cantidad de información representativa posible, descartándose la mayor cantidad de información redundante, con el objetivo de disminuir el tiempo requerido para su visualización. La segunda se refiere a que el resumen no solamente debe ser informativo, sino que debe poseer características visuales agradables para el usuario que finalmente lo visualiza [Truong y Venkatesh, 2007, Ren y otros, 2010, Herranz y Martínez, 2010]. En el contexto de la investigación la usabilidad se refiere, además, a que los resúmenes generados permitan determinar el contenido del video original, para describirlo y poder tomar decisiones sobre el mismo.
Para validar la usabilidad del resumen se comprueban los niveles de escalabilidad, de información y de síntesis representativa del método propuesto en el capítulo 2.

- Escalabilidad: relativo a las condiciones para lograr distintos resúmenes de la misma secuencia variando su longitud.
- Información: que se refiere a la capacidad para lograr que el usuario final obtenga una comprensión global de la información contenida en el video original.
- Síntesis representativa: referido a las potencialidades para proporcionar una síntesis de la información del video original haciendo prevalecer el contenido representativo.

Para validar el cumplimiento de estas condiciones por el método propuesto se aplicó un instrumento de diagnóstico. Esta forma de validación para resúmenes automáticos de video no es no-

vedosa, en aproximaciones anteriores [Ngo y otros, 2003, Santini, 2007, De Avila y otros, 2008, Herranz y Martínez, 2010] se observan variantes similares. Se utiliza una técnica de muestreo aleatorio simple considerando la población y muestra que se caracteriza a continuación.

3.3.1. Caracterización de la población y muestra

La población definida, está constituida por un grupo de 40 personas que son usuarios de las aplicaciones de gestión, procesamiento y transmisión de contenidos audiovisuales desarrolladas en GEYSED (en este caso de AGORAV y PRIMICIA). También se incluyen 15 especialistas que actualmente se desempeñan como investigadores y desarrolladores de las nuevas versiones de las aplicaciones (AGORAV, PRIMICIA y el STCV); así como cinco especialistas que laboran en medios de comunicación audiovisual para un total de 60 personas.

Para determinar el tamaño de la muestra se sigue el procedimiento propuesto en [Hernández Sampieri y otros, 2010] con un nivel de confianza del 95 %, resultando de 23 personas. Con el objetivo de lograr una representatividad en la muestra equivalente a la población se toman 15 usuarios, seis especialistas investigadores y desarrolladores y dos especialistas de los medios de comunicación.

3.3.2. Aplicación del instrumento de diagnóstico

El instrumento aplicado es una encuesta, ver Anexo 3.7, a las personas que conforman la muestra. La encuesta consta de cuatro preguntas en las que se pretende diagnosticar el grado de usabilidad, según los niveles antes expuestos, de los resúmenes resultantes al aplicar el método propuesto en el capítulo 2.

La evaluación mediante una encuesta puede poseer cierto grado de subjetividad ya que las respuestas al instrumento aplicado pueden tener determinado nivel de incertidumbre, es decir, no necesariamente se obtendrá una valoración fiel de los resúmenes generados por parte de los encuestados, por esta razón en el instrumento se solicita asignar un valor real y dos relativos, determinados por el valor mínimo y máximo que podría alcanzar respectivamente. Se concibe que asignen un número n para cada pregunta, $n \in \mathbb{N}\ [0, 10]$, estableciendo los tres valores que determinan su criterio sobre la usabilidad de los resúmenes generados.

Debido a que se tiene en cuenta el grado de subjetividad evidente en este contexto, para obte-

ner resultados más certeros se realiza el análisis basado en la teoría de los conjuntos borrosos [Zadeh, 1965, Zadeh, 1975]. Específicamente se utiliza el método de números borrosos triangulares propuestos en la investigación de [Yager, 1988] y que se ha utilizado por [Jiménez Moya, 2013].

DEFINICIÓN 7 *Un número borroso triangular se define como un número impreciso caracterizado por tres valores* $\overline{A} = [a_1; a_2; a_3]$. *Cumpliéndose la condición* $a_1 \leq a_2 \leq a_3$.

- *El valor central* a_2 *constituye el valor de mayor nivel de confianza.*
- *Los valores* a_1, a_3 *constituyen los límites inferior y superior; expresan los valores de incertidumbre y no poseen nivel de confianza.*

La aritmética sobre números borrosos triangulares es equivalente, solamente se deben tener en cuenta las peculiaridades de estos números.
Para procesar estadísticamente los valores borrosos obtenidos al aplicar el instrumento de diagnóstico a la muestra, se calculó la media $\overline{X}$, lo que se puede denominar como media borrosa. La formulación para determinarla, se obtiene adaptando la fórmula clásica [Hernández Sampieri y otros, 2010] para hallar la media sobre el conjunto borroso como se expresa seguidamente en 3.3.

$$\overline{X} = [\frac{\sum_1^n a_1}{n}, \frac{\sum_1^n a_2}{n}, \frac{\sum_1^n a_3}{n}] \tag{3.3}$$

Los resultados de la encuesta realizada se pueden observar en la tabla 3.4. En la última fila se expresa la media borrosa de la muestra, determinada a partir de las consideraciones de los encuestados.

Tabla 3.4: Resultados de la aplicación del instrumento de diagnóstico

Encuestados	**Pregunta 1**			**Pregunta 2**			**Pregunta 3**			**Pregunta 4**		
1	4	6	7	3	5	7	6	7	8	5	5	6
2	5	7	8	4	6	7	7	8	9	4	6	8
3	6	8	9	7	8	8	6	7	8	6	8	10
4	4	5	6	5	6	7	5	5	5	5	6	7
5	6	7	8	4	6	8	4	5	6	7	8	9
6	5	6	7	5	7	9	5	6	7	6	7	8
7	7	7	8	4	6	7	4	5	6	6	7	8
8	5	6	7	7	8	8	6	7	8	7	8	9
9	7	8	9	7	9	9	5	7	9	7	8	9
10	5	7	8	6	8	9	4	6	8	6	7	8
11	7	8	9	6	7	8	5	7	9	7	8	9
12	4	5	6	5	6	7	4	6	8	4	5	6
13	5	6	7	4	5	6	6	7	8	4	6	8
14	7	8	9	5	7	9	6	8	10	6	8	10
15	7	8	9	4	7	9	4	6	8	5	6	7
16	5	7	9	7	8	9	5	7	9	7	8	9
17	7	8	9	6	7	8	6	8	10	6	8	10
18	6	8	9	7	8	9	5	7	9	7	8	9
19	5	7	8	6	8	8	5	7	9	4	6	8
20	8	8	9	6	7	8	6	8	10	5	7	9
21	6	7	8	5	6	7	5	8	9	5	6	7
22	7	8	8	5	7	9	5	6	7	7	8	9
23	6	7	8	5	6	8	4	5	6	6	7	8
Resultados												
$\overline{X}$	5,83	7,04	8,04	5,35	6,87	8	5,13	6,65	8,09	5,74	7	8,3

Los valores centrales obtenidos se han validado para verificar la consistencia y veracidad del instrumento de diagnóstico aplicado. El método utilizado fue el Alpha de Cronbach [Cronbach, 1951]. Se ha realizado procesando, con el software R versión 3.0.2, el conjunto de valores centrales resultantes [Horton y Kleinman, 2010]. Se obtuvo un $\alpha = 0{,}724$ que se puede considerar aceptable [Peterson, 1994, Gliem y Gliem, 2003].

Tabla 3.5: Normalización de los resultados del diagnóstico

$\overline{X}$	0,58	0,7	0,8	0,54	0,69	0,8	0,51	0,67	0,81	0,57	0,7	0,83

Para valorar los resultados obtenidos se establece una escala haciendo corresponder los valores numéricos hallados con valores nominales.

Tabla 3.6: Escala Nominal-Numérico

Escala	
Valor nominal	**Valor numérico**
Muy bajo	$0 < \overline{X} \leq 0{,}2$
Bajo	$0{,}2 < \overline{X} \leq 0{,}4$
Medio	$0{,}4 < \overline{X} \leq 0{,}6$
Alto	$0{,}6 < \overline{X} \leq 0{,}8$
Muy alto	$0{,}8 < \overline{X} \leq 1$

Como se puede apreciar en los resultados, se obtiene una valoración donde los valores mínimos de la media borrosa se ubican en el nivel Medio de la escala definida, pero con tendencia al nivel Alto; los valores centrales de la terna, que poseen mayor nivel de confianza, se ubican en el nivel Alto de la escala y los valores máximos de la terna se ubican en el nivel Muy Alto. A consideración del autor de la investigación, se puede asumir que la usabilidad de los resúmenes generados es Alta.

3.4. Compatibilidad del método con distintas codificaciones

Una de las consideraciones principales por las que se ha desarrollado la investigación, lo constituye que aproximaciones propuestas en la literatura no concebían la obtención de resúmenes para estándares que no fueran los definidos por el MPEG. Por esta razón en este epígrafe se ha comprobado el comportamiento del método sobre secuencias de videos en distintas codificaciones como se muestra en la tabla 3.7. Las seis secuencias han sido seleccionadas por el autor aleatoriamente en un grupo de videos, intentando representar las codificaciones que ha considerado más populares y utilizadas.
Como se evidencia en los resultados con el método propuesto en la investigación se pueden procesar secuencias con diversos estándares de codificación.

Tabla 3.7: Resultado de las pruebas para verificar la compatibilidad con varias codificaciones

Secuencia	**Contenedor**	**Codificador**	**Tasa de bits (Kbps)**	**Fotogramas por Segundo)**	**Resolución**	**Resultado**
Stay	AVI	MPEG4-Xvid	1280	29	800X600	Procesado correcta-mente
Girl	WMV	WMV2	2400	30	720X480	Procesado correcta-mente
Cinema	MPEG	MPEG-1	1150	25	720X480	Procesado correcta-mente
Been	OGG[2]	Theora	1590	29	720X480	Procesado correcta-mente
Dance	WEBM[3]	VP-8	1590	29	720X480	Procesado correcta-mente
Fire	MKV	MPEG4-(H264)	1536	25	1280x538	Procesado correcta-mente

3.5. Comparación de los resultados obtenidos con otra aproximación

En el trabajo [Herranz y Martínez, 2010] que se ha tomado como base para el método propuesto, se presentan los resultados en cuanto al tiempo, en segundos, requerido para la obtención de resúmenes de distintas escalas y modalidades. Dicha investigación solamente comprueba el tiempo de procesamiento sobre un video del que solamente expresa que posee una duración de 10 minutos. En la tabla 3.8 se muestran los resultados de la investigación antes mencionada.

Tabla 3.8: Tiempo de procesamiento (en segundos) para una secuencia de diez minutos, fuente [Herranz y Martínez, 2010].

Análisis	**Generación**							
(-)	**Resúmen estático (*N*º)**		**Resúmen dinámico (%)**					
(-)	5	30	1	5	10	20	50	100
2,21	0,64	0,68	0,68	0,70	0,79	1,46	7,09	16,90

Los videos utilizados han sido seleccionados de forma aleatoria por el autor considerando que su duración fuera de 10 minutos y sus propiedades se pueden visualizar en la tabla 3.9.

Tabla 3.9: Propiedades de las secuencias seleccionadas para comprobar el rendimiento

Archivo de video	Referencia	Duración	Fotogramas por segundo	Resolución	Contenedor	Codificación
Harry	V-6	10	29,97	1920x1080	MKV	H264
Break	V-7	10	30	1280x720	WMV	WMV2
Phone	V-8	10	29,97	720X480	WEBM	Theora
Nature	V-9	10	25	352X288	MPEG	MPEG
Beast	V-10	10	29,97	1280x720	OGG	VP8

Tabla 3.10: Tiempo de procesamiento (en segundos) del método propuesto

Referencia	**Análisis**	**Generación**							
(-)	(-)	Resúmen estático (N^o)		Resúmen dinámico (%)					
(-)	(-)	5	30	1	5	10	20	50	100
V-6	6,73	1,04	1,17	3,01	3,42	4,29	5,04	8,06	18,74
V-7	7,06	1,11	1,49	2,83	3,37	4,13	4,57	7,83	17,02
V-8	6,22	1,02	1,58	2,68	3,14	3,59	4,36	7,49	17,03
V-9	3,22	0,89	0,94	1,04	1,69	2,72	3,86	7,10	17,01
V-10	6,81	1,34	1,71	2,63	3,02	3,79	4,46	7,46	16,98

En la tabla 3.10 se muestran los resultados de las pruebas de rendimiento. Es posible observar que el método propuesto posee un rendimiento inferior al que propone [Herranz y Martínez, 2010] para secuencias de duración igual a diez minutos. La diferencia principal, se visualiza en la etapa de análisis, evidenciándose la superioridad de la propuesta de [Herranz y Martínez, 2010]. En la etapa de generación para lograr los resúmenes de mayor escala el rendimiento es bastante similar, aunque, también el de [Herranz y Martínez, 2010] muestra superioridad para resúmenes de menor escala. No obstante, en el procesamiento influyen otras variables como pueden ser, la resolución de la secuencia original, la cantidad de fotogramas por segundo, así como la cantidad de tomas, que no se recogen en la investigación, por lo que no es posible determinar las características del video analizado en [Herranz y Martínez, 2010]; solamente se publica que posee una duración de diez minutos. En la secuencia V-9 que posee la menor resolución y cantidad de fotogramas por segundo, además codificada bajo estándares del MPEG, se muestran los mejores índices de rendimiento.

Por otra parte en la investigación de [Herranz y Martínez, 2010] se computa únicamente el descriptor *color layout*. En el caso del método propuesto, como se explicó anteriormente, se computan los descriptores de color y bordes, lo que permite obtener más información y lograr los índices de *Precision-Recall* que se exponen en el epígrafe 3.2. Otro elemento que influye, es la utilización de estándares del MPEG y el hecho de que en [Herranz y Martínez, 2010] se asuman los GOPs como unidad básica de procesamiento, por lo que se omite el procedimiento para determinar los límites entre tomas. Además utilizan como fotograma principal el (I) de cada GOP, por lo que tampoco realizan procesamiento para establecer el fotograma principal. Precisamente este último elemento es lo que limita la propuesta de [Herranz y Martínez, 2010] para que se pueda utilizar en audiovisuales que no estén codificados bajo estándares del MPEG.

Los resultados demuestran que se debe continuar perfeccionando el método propuesto, pues aunque se logran los resúmenes, el rendimiento no supera los resultados de [Herranz y Martínez, 2010]. A pesar de lo anterior, el método, como se explica en el epígrafe 3.6, no concibe la etapa de análisis con intervención del usuario. Esta etapa se ejecutará automáticamente al realizar la gestión de un audiovisual por lo que no es alarmante el tiempo dedicado.

3.6. Integración del método en aplicaciones desarrolladas en GEYSED

En las aplicaciones de gestión, procesamiento y transmisión de contenidos audiovisuales desarrolladas en GEYSED, se utiliza una arquitectura que posee un componente denominado gestor de procesos [Díaz y otros, 2013] cuya finalidad es administrar los recursos existentes para garantizar que se ejecuten exitosamente todos los procesos que se requieren en las aplicaciones. El gestor a su vez, adiciona nuevos procesos mediante la creación de *plugins* con una estructura definida y una vez incluidos, posteriormente pueden ser gestionados [Suárez Pérez y otros, 2012]. Como parte de estos procesos actualmente el gestor cuenta con los de codificación, ingesta e indexación de video.

En el momento que se escribe esta memoria, no se ha integrado el método a las aplicaciones antes mencionadas pero se hará referencia a las consideraciones necesarias para lograrlo. La integración del método propuesto 2 se debe hacer desarrollando dos *plugins*.

- Uno de los *plugins* se encargaría exclusivamente de la etapa de análisis, debido a que esta etapa se va a realizar automáticamente una vez que se comience la gestión de un material audiovisual y no es necesario que el usuario intervenga en su ejecución, solamente se deben gestionar los errores en caso de existir. Una vez ejecutado el análisis quedan listos los datos obteniéndose la representación del contenido del video que garantizará la etapa de generación.

- Por su parte el otro *plugin* se encargaría de la etapa de generación. Para esto recibe como parámetros de entrada la escala necesaria y la modalidad de resumen requerida, el resto de los datos necesarios se obtuvieron previamente al realizar la etapa de análisis. El gestor comienza la etapa de generación cuando reciba una petición de resumen, se validan los errores que puedan exitir y en caso de que el resultado sea el esperado, se muestra al usuario el resumen obtenido según los requisitos de escalabilidad y la modalidad seleccionada.

3.7. Conclusiones parciales

- La medición de *precision-recall* obteniendo valores por encima de 0,86 evidencia que se garantiza la eficacia de la segmentación en la etapa de análisis.

- La encuesta realizada para comprobar la usabilidad del método propuesto para generar automáticamente resúmenes de video arrojó resultados satisfactorios, demostrando la escalabilidad y usabilidad del mismo.

- Las pruebas de compatibilidad con codificaciones evidencian que es posible utilizar el método en videos con distinta codificación.

- Las pruebas de rendimiento demostraron que el método no posee mejor rendimiento que otros existentes en la literatura. Sin embargo, el rendimiento para escalas superiores, secuencias de baja resolución y tasa de bits es similar.

Conclusiones

- El estudio de referentes teóricos en la literatura evidenció las tendencias para la creación de resúmenes de video, las mismas se aplicaron en la concepción del método propuesto.

- El método propuesto en la investigación permite generar resúmenes escalables de secuencias de video, lo que favorece la navegación sobre las mismas y facilita la descripción de su contenido así como la toma de decisiones.

- Las pruebas realizadas al método propuesto evidencian su efectividad, pero establecen la necesidad de continuar trabajando en esta temática para lograr índices de rendimiento superiores.

- El método propuesto se puede integrar sin dificultad en aplicaciones de gestión, procesamiento y transmisión de archivos audiovisuales desarrolladas en el centro GEYSED de la UCI.

Recomendaciones

- Incorporar técnicas de clasificación al método propuesto para lograr resúmenes que se basen en conceptos semánticos presentes en el contenido del video.
- Integrar el método propuesto en las aplicaciones de gestión, procesamiento y transmisión de archivos audiovisuales desarrollados en el centro GEYSED de la UCI.

Publicaciones relacionadas del autor

Durante la investigación llevada a cabo, el autor estuvo involucrado en publicaciones relacionadas:

Memorias en congresos y talleres:

- VI Congreso Internacional de Tecnologías, Contenidos Multimedia y Realidad Virtual, XV Convención y Feria Internacional Informática 2013. Autor del trabajo "Sistema de Gestión y Transmisión de Contenidos Audiovisuales".
- VI Congreso Internacional de Tecnologías, Contenidos Multimedia y Realidad Virtual, XV Convención y Feria Internacional Informática 2013. Co-autor del trabajo "Plataforma para la aplicación de los Set-Top Box en Cuba, un avance social".
- VI Congreso Internacional de Tecnologías, Contenidos Multimedia y Realidad Virtual, XV Convención y Feria Internacional Informática 2013. Co-autor del trabajo "Subsistema de consulta, visualización y descarga de archivos multimedia".

Artículos en revistas:

- Revista Cubana de Ciencias Informáticas, 2014. Autor del artículo "Generación de resúmenes escalables de video".
- Revista Digital Sociedad de la Información 2012. Co-autor del artículo "Descriptores de video, sus aplicaciones en materiales audiovisuales".
- Revista de Informática Educativa y Medios Audiovisuales, 2011. Co-autor del artículo "Diseño de la base de datos para sistemas de digitalización y gestión de medias".

Referencias bibliográficas

[DRA, 2001] (2001). Diccionario de la lengua española (DRAE).

[Acevedo Martínez, 2013] Acevedo Martínez, L. (2013). Software de sistemas: Software para sistemas concurrentes y paralelos.

[Acharya y Ray, 2005] Acharya, T. y Ray, A. K. (2005). *Image processing: principles and applications*. John Wiley & Sons.

[Alexandrescu, 2001] Alexandrescu, A. (2001). *Modern C++ design: generic programming and design patterns applied*. Addison-Wesley.

[Ali y Clausi, 2001] Ali, M. y Clausi, D. (2001). Using the canny edge detector for feature extraction and enhancement of remote sensing images. En *Geoscience and Remote Sensing Symposium, 2001. IGARSS'01. IEEE 2001 International*, volumen 5, pp. 2298–2300. IEEE.

[Amiri y otros, 2011] Amiri, A., Abdollahi, N., Jafari, M., y Fathy, M. (2011). Hierarchical keyframe based video shot clustering using generalized trace kernel. En Pichappan, P., Ahmadi, H., y Ariwa, E., editores, *Communications in Computer and Information Science*, volumen 241, pp. 251–257. Springer Berlin Heidelberg.

[Arbelaez y otros, 2009] Arbelaez, P., Maire, M., Fowlkes, C., y Malik, J. (2009). From contours to regions: An empirical evaluation. En *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 2294–2301. IEEE.

[Arco García, 2008] Arco García, L. (2008). *Agrupamiento basado en la intermediación diferencial y su valoración utilizando la teoría de los conjuntos aproximados*. Tesis doctoral, UNIVERSIDAD CENTRAL"MARTA ABREU" DE LAS VILLAS.

[Basseville, 1989] Basseville, M. (1989). Distance measures for signal processing and pattern recognition. *Signal processing*, 18(4):349–369.

[Bescós, 2003] Bescós, J. (2003). Tr-2003/06, "shot tansitions gorund truth for the mpeg content set". Technical report, Universidad Autónoma de Madrid.

[Bescós y otros, 2005] Bescós, J., Cisneros, G., Martínez, J. M., Menéndez, J. M., y Cabrera, J. (2005). A unified model for techniques on video-shot transition detection. *IEEE TRANSACTIONS ON MULTIMEDIA*, 7(2):–.

[Borth y otros, 2008] Borth, D., Ulges, A., Schulze, C., y Breuel, T. M. (2008). Keyframe extraction for video tagging & summarization. En *Informatiktage*, volumen 2008, pp. 45–48.

[Bosch y otros, 2007] Bosch, A., Zisserman, A., y Munoz, X. (2007). Representing shape with a spatial pyramid kernel. En *Proceedings of the 6th ACM international conference on Image and video retrieval*, pp. 401–408. ACM.

[Boullosa García, 2011] Boullosa García, s. (2011). Estudio comparativo de descriptores visuales para la detección de escenas cuasi-duplicadas.

[Boyle y Thomas, 1988] Boyle, R. y Thomas, R. (1988). *Computer vision: a first course*. Artificial intelligence texts. Blackwell Scientific Publications.

[Bradski y Kaehler, 2008] Bradski, G. y Kaehler, A. (2008). *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media.

[Bregonzio y otros, 2009] Bregonzio, M., Gong, S., y Xiang, T. (2009). Recognising action as clouds of space-time interest points. pp. 1948–1955.

[Brutzer y otros, 2011] Brutzer, S., Hoferlin, B., y Heidemann, G. (2011). Evaluation of background subtraction techniques for video surveillance. En *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 1937–1944. IEEE.

[Canny, 1986] Canny, J. (1986). A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):679–698.

[Castellanos y otros, 2010] Castellanos, R., Kalva, H., Marques, O., y Furht, B. (2010). Event detection in video using motion analysis. En *Proceedings of the first ACM international workshop on Analysis and retrieval of tracked events and motion in imagery streams*, pp. 57–62. ACM.

[Chan y otros, 2011] Chan, P. P. K., Hui, Y., Ng, W. W. Y., y Yeung, D. S. (2011). A novel method to reduce redundancy in adaptive threshold clustering key frame extraction systems.

En *Machine Learning and Cybernetics (ICMLC), 2011 International Conference on*, volumen 4, pp. 1637–1642.

[Chandrakar y Bhonsle, 2012] Chandrakar, N. y Bhonsle, D. (2012). Study and comparison of various image edge detection techniques. *International Journal of Managment, IT and Engineering*, 2(5):499–509.

[Chavan y otros, 2013] Chavan, S. A., Telang, R. B., y Akojwar, S. G. (2013). A review on cooperative shot boundary detection. En India, C. S. o., editor, *International Conference on Recent Trends and Innovations in Engineering & Technology*, pp. –.

[Chen y otros, 2011] Chen, S.-C., Shyu, M.-L., Chen, C., y Liu, D. (2011). Within and between shot information utilisation in video key frame extraction. 10(03):247–259.

[Choi y otros, 2010] Choi, S.-S., Cha, S.-H., y Tappert, C. C. (2010). A survey of binary similarity and distance measures. *Journal of Systemics, Cybernetics and Informatics*, 8(1):43–48.

[CISCO, 2012] CISCO (2012). Cisco visual networking index: Forecast and methodology, 2011-2016. Technical report, CISCO.

[Cronbach, 1951] Cronbach, L. J. (1951). Coefficient alpha and the internal structure of tests. *psychometrika*, 16(3):297–334.

[Cózar, 2010] Cózar, J. R. (2010). Notas del curso técnicas de reconocimiento de patrones para el análisis del contenido de vídeo digital impartido por el dr. julián ramos cózar.

[Dan y otros, 2011] Dan, R., Olsen, J., y Brandon, M. (2011). Video summarization based on user interaction. En *EuroITV '11, 11th European Conference on Interactive TV*, pp. 115–122, Lisboa, Portugal. ACM.

[Davies, 2012] Davies, E. R. (2012). *Computer and machine vision: theory, algorithms, practicalities*. Academic Press.

[Díaz Ales y Alonso Guerrero, 2012] Díaz Ales, N. T. y Alonso Guerrero, Z. (2012). Agoritmo para el reconocimiento de rostro en sistemas de catalogación de audiovisuales. En *Festival Internacional de Radio y TV*.

[Díaz Berenguer y otros, 2013] Díaz Berenguer, A., Pacheco Jérez, Y. S., Chávez Ayala, D., Navarro Sánchez, A., y Quintana Rondón, Y. (2013). Sistema de gestión y transmisión de contenidos audiovisuales. En *"VI Congreso Internacional de Tecnologías, Contenidos Multimedia y Realidad Virtual". Informática 2013.*

[Díaz Espinoza, 2011] Díaz Espinoza, D. A. (2011). Implementación y comparación de descriptores para búsquedas en video.

[De Avila y otros, 2008] De Avila, S. E. F., , Da Luz, A. J., , A. A. d. A., y Cord, M. (2008). VSUMM: An approach for automatic video summarization and quantitative evaluation. En *XXI Brazilian Symposium on Computer Graphics and Image Processing*. IEEE Comptuter Society.

[Deza y Deza, 2012] Deza, M. y Deza, E. (2012). *Encyclopedia of Distances*. SpringerLink : Bücher. Springer.

[Díaz y otros, 2013] Díaz, A. F., Pérez, J. M. S., y Torreira, Y. B. (2013). Procesamiento audiovisual distribuido en entornos de alta demanda computacional. *Serie Científica*, 6(7).

[Díaz Berenguer, 2014] Díaz Berenguer, A. (2014). Generación de resúmenes escalables de video. *Revista Cubana de Ciencias Informáticas*, 8(1):33–41.

[Ding y Goshtasby, 2001] Ding, L. y Goshtasby, A. (2001). On the canny edge detector. *Pattern Recognition*, 34(3):721–725.

[Don y Uma, 2009] Don, A. y Uma, K. (2009). Adaptive edge-oriented shot boundary detection.

[Dumont y Mérialdo, 2008] Dumont, E. y Mérialdo, B. (2008). Redundancy removing by adaptive acceleration and event clustering for video summarization. En *Ninth International Workshop on Image Analysis for Multimedia Interactive Services*, Klagenfurt University, Austria. IEEE Comptuter Society.

[Dunn y Everitt, 2004] Dunn, G. y Everitt, B. S. (2004). *An introduction to mathematical taxonomy*. Courier Dover Publications.

[Ejaz y otros, 2012] Ejaz, N., Tariq, T. B., y Baik, S. W. (2012). Adaptive key frame extraction for video summarization using an aggregation mechanism. *Journal of Visual Communication & Image Representation*, 23(7):1031–1040.

[Emna Fendri, 2010] Emna Fendri, Hanene Ben-Abdallah, A. B. H. (2010). A novel approach for soccer video summarization. En *Second International Conference on MultiMedia and Information Technology*. IEEE.

[Everitt y otros, 2001] Everitt, B., Landau, S., y Leese, M. (2001). Cluster analysis arnold. *A member of the Hodder Headline Group, London*.

[Gliem y Gliem, 2003] Gliem, J. A. y Gliem, R. R. (2003). Calculating, interpreting, and reporting cronbach's alpha reliability coefficient for likert-type scales. Midwest Research-to-Practice Conference in Adult, Continuing, and Community Education.

[Gómez Carranza y Moens, 2014] Gómez Carranza, J. C. y Moens, M.-F. (2014). Text based information retrieval.

[González y Woods, 2002] González, R. C. y Woods, R. E. (2002). *Digital Image Processing*. Prentice Hall, second edición.

[Gove, 2010] Gove, D. (2010). *Multicore Application Programming: for Windows, Linux, and Oracle, Solaris*. Addison Wesley.

[Hampapur y otros, 2012] Hampapur, A., Gorkani, M., Shu, C., y Gupta, A. (2012). Key frame selection.

[Hastie y otros, 2009] Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Friedman, J., y Tibshirani, R. (2009). *The elements of statistical learning*, volumen 2. Springer.

[Hernández García y otros, 2010] Hernández García, R., Montaner Hernández, Y., Hernández Bustio, J. A., y Olivares Tamayo, J. D. (2010). Primicia, plataforma de televisión informativa. *Revista de Ingeniería Electrónica, Automática y Comunicaciones*, 1(2):45–50.

[Hernandez Heredia, 2013] Hernandez Heredia, Y. (2013). *MODELO PARA LA DETECCIÓN Y RECONOCIMIENTO DE ACCIONES HUMANAS EN VIDEOS A PARTIR DE DES-*

CRIPTORES ESPACIO-TEMPORALES. Tesis doctoral, Universidad de las Ciencias Informáticas. UCI.

[Hernandez Heredia y otros, 2012] Hernandez Heredia, Y., Ortiz Rojas, J., Hernández García, R., y González Linares, J. M. (2012). Descriptores temporales-espaciales en la detección automática de información audiovisual. *Ciencias de la Información*, 3(2):21–27.

[Hernández Sampieri y otros, 2010] Hernández Sampieri, R., Fernández Collado, C., y Baptista Lucio, P. (2010). *Metodología de la investigación*. México: Editorial Mc Graw Hill, 5ta edición.

[Hernández Heredia, 2010] Hernández Heredia, Y. (2010). Metodología para la detección de objetos en sistemas para la catalogación semi-automática y automática de videos. Tesis de máster, Universidad de las Ciencias Informáticas.

[Herranz y otros, 2012] Herranz, L., Calic, J., Martínez, J. M., y Mrak, M. (2012). Scalable comic-like video summaries and layout disturbance. *Multimedia, IEEE Transactions on*, 14(4):1290–1297.

[Herranz y Martínez, 2010] Herranz, L. y Martínez, J. M. (2010). A framework for scalable summarization of video. *Circuits and Systems for Video Technology, IEEE Transactions on*, 20(9):1265–1270.

[Herranz Arribas, 2010] Herranz Arribas, L. (2010). *A Scalable Approach to Video Summarization and Adaptation*. Tesis doctoral, Universidad Autónoma de Madrid. Escuela Politécnica Superior de Ingeniería Informática, Madrid.

[Horton y Kleinman, 2010] Horton, N. y Kleinman, K. (2010). *Using R for Data Management, Statistical Analysis, and Graphics*. Using R for Data Management, Statistical Analysis, and Graphics. Taylor & Francis.

[Jain y otros, 1999] Jain, A. K., Murty, M. N., y Flynn, P. J. (1999). Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323.

[Jalab, 2011] Jalab, H. A. (2011). Image retrieval system based on color layout descriptor and gabor filters. En *Open Systems (ICOS), 2011 IEEE Conference on*, pp. 32–36. IEEE.

[Jiang y otros, 2013] Jiang, X., Sun, T., Liu, J., Chao, J., y Zhang, W. (2013). An adaptive video shot segmentation scheme based on dual-detection model. *Neurocomputing*, 116:102–111.

[Jiménez Moya, 2013] Jiménez Moya, G. E. (2013). Extensiones para el control de la ejecución de proyectos basadas en el análisis de la dimensión geográfica. Tesis de máster, Universidad de las Ciencias Informáticas UCI.

[Jinhui y otros, 2007] Jinhui, Y., Huiyi, W., Lan, X., Wujie, Z., Jianmin, L., Fuzong, L., y Bo, Z. (2007). A formal study of shot boundary detection. En *Circuits and Systems for Video Technology, IEEE Transactions on*, volumen 17, pp. 168–186. IEEE.

[Khan Gramsci, 2011] Khan Gramsci, S. (2011). *A Scalable Video Streaming Approach using Distributed B-Tree*. Tesis doctoral, THE UNIVERSITY OF BRITISH COLUMBIA, Bacouver.

[Kim y Hwang, 2002] Kim, C. y Hwang, J.-N. (2002). Object-based video abstraction for video surveillance systems. *Circuits and Systems for Video Technology, IEEE Transactions on*, 12(12):1128–1138.

[Klicnar y Beran, 2012] Klicnar, L. y Beran, V. (2012). Robust motion segmentation for on-line application.

[Laganière, 2011] Laganière, R. (2011). *OpenCV 2 Computer Vision Application Programming Cookbook: Over 50 recipes to master this library of programming functions for real-time computer vision*. Packt Publishing Ltd.

[Laganière, 2014] Laganière, R. (2014). *OpenCV Computer Vision Application Programming Cookbook Second Edition*. Packt Publishing Ltd.

[Laptev, 2005] Laptev, I. (2005). On space-time interest points. *International Journal of Computer Vision*, 64(2-3):107–123.

[Lee y otros, 2011a] Lee, H., Yu, J., Im, Y., Gil, J.-M., y Park, D. (2011a). A unified scheme of shot boundary detection and anchor shot detection in news video story parsing. 51(3):1127–1145.

[Lee y otros, 2011b] Lee, Y. J., Kim, J., y Grauman, K. (2011b). Key-segments for video object segmentation. En *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 1995–2002. IEEE.

[Lienhart, 2001] Lienhart, R. (2001). Reliable transition detection in videos: A survey and practitioners guide. En *International Journal of Image and Graphics*, volumen 1.

[Lienhart, 1998] Lienhart, R. W. (1998). Comparison of automatic shot boundary detection algorithms. En *Electronic Imaging'99*, pp. 290–301. International Society for Optics and Photonics.

[Liping y otros, 2010] Liping, R., Zhiyi, Q., Weiqin, N., Chaoxin, N., y Yanqiu, C. (2010). Key frame extraction based on information entropy and edge matching rate. En *Future Computer and Communication (ICFCC), 2010 2nd International Conference on*, volumen 3, pp. V3–91–V3–94–.

[Liu y Yang, 2013] Liu, G.-H. y Yang, J.-Y. (2013). Content-based image retrieval using color difference histogram. *Pattern Recognition*, 46(1):188–198.

[López Vidales y otros, 2011] López Vidales, N., González Aldea, P., y Medina de la Viña, E. (2011). Jóvenes y televisión en 2010 un cambio de hábitos. 30:97–113.

[Lu y Shi, 2013] Lu, Z. y Shi, Y. (2013). Fast video shot boundary detection based on svd and pattern matching. pp. –.

[Lupatini y otros, 1998] Lupatini, G., Saraceno, C., y Leonardi, R. (1998). Scene break detection: a comparison. En *Research Issues In Data Engineering, 1998. 'Continuous-Media Databases and Applications'. Proceedings., Eighth International Workshop on*, pp. 34–41–.

[Lux y otros, 2007] Lux, M., Schàffmann, K., Marques, O., y Bàszàrmenyi, L. (2007). A novel tool for quick video summarization using keyframe extraction techniques. Technical report, Institute for Information Technology Klagenfurt University and Department of Computer Science and Engineering Atlantic University.

[Maass y González, 2005] Maass, M. y González, J. A. (2005). De memorias y tecnologías: radio, televisión e internet en méxico. *Estudios sobre las culturas contemporáneas*, 11(22).

[McIlhagga, 2011] McIlhagga, W. (2011). The canny edge detector revisited. *International Journal of Computer Vision*, 91(3):251–261.

[Mendi y Bayrak, 2010] Mendi, E. y Bayrak, C. (2010). Shot boundary detection and key frame extraction using salient region detection and structural similarity. En *Proceedings of the 48th Annual Southeast Regional Conference*, pp. 1–4, Oxford, Mississippi. ACM.

[Mitchell y otros, 1996] Mitchell, J. L., Pennebaker, W. B., Fogg, C. E., y LeGall, D. J. (1996). *MPEG Video Compression Standard.* Kluwer Academic, Nueva York, EE.UU.

[Mohanta y otros, 2010] Mohanta, P. P., Saha, S. K., y Chanda, B. (2010). A heuristic algorithm for video scene detection using shot cluster sequence analysis. En *Proceedings of the Seventh Indian Conference on Computer Vision, Graphics and Image Processing*, pp. 464–471–, Chennai, India. ACM.

[Mohanty y Kanungo, 2013] Mohanty, K. y Kanungo, P. (2013). Automatic cut detection based video segmentation. En *Computational and Business Intelligence (ISCBI), 2013 International Symposium on*, pp. 265–268. IEEE.

[Ngo y otros, 2003] Ngo, C.-W., Ma, Y.-F., y Zhang, H.-J. (2003). Automatic video summarization by graph modeling. En *Ninth IEEE International Conference on Computer Vision (ICCV 2003)*. IEE.

[Over y otros, 2008] Over, P., Smeaton, A. F., y Awad, G. (2008). The trecvid 2008 bbc rushes summarization evaluation. En *Proceedings of the 2nd ACM TRECVid Video Summarization Workshop*, pp. 1–20. ACM.

[Panetta y otros, 2011] Panetta, K. A., Agaian, S. S., Nercessian, S. C., y Almunstashri, A. A. (2011). Shape-dependent canny edge detector. *Optical Engineering*, 50(8):087008–087008.

[Parry y otros, 2011] Parry, M. L., Legg, P. A., Chung, D. H. S., Griffiths, I. W., y Chen, M. (2011). Hierarchical event selection for video storyboards with a case study on snooker video visualization. 17(12):1747–1756.

[Peterson, 1994] Peterson, R. A. (1994). A meta-analysis of cronbach's coefficient alpha. *Journal of consumer research*, pp. 381–391.

[Pickering y Rüger, 2003] Pickering, M. J. y Rüger, S. (2003). Evaluation of key frame-based retrieval techniques for video. *Computer Vision and Image Understanding*, 92:217–235.

[Powers, 2011] Powers, D. M. (2011). Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation.

[PRASANNA, 2013] PRASANNA, S. (2013). *INTELLIGENT MULTIMODEL CONTENT BASED VIDEO RETRIEVAL*. Tesis doctoral, VELS University. Institute of Sciences, Technology and Advanced Studies (VISTAS). Chenai. India.

[Pérez Suárez y otros, 2008] Pérez Suárez, A., García Delgado, G., Medina Pagola, J. E., Martínez Trinidad, J. F., y Carrasco Ochoa, J. A. (2008). Algoritmos de agrupamiento para colecciones de documentos. Technical report, Centro de Aplicaciones de Tecnologías Avanzadas CENATAV. Serie Gris.

[Qifan y otros, 2013] Qifan, F., Yichun, Z., Liyong, X., y Huixin, L. (2013). A method of shot-boundary detection based on hsv space. En *Computational Intelligence and Security (CIS), 2013 9th International Conference on*, pp. 219–223.

[Ren y otros, 2010] Ren, J., Jiang, J., y Feng, Y. (2010). Activity-driven content adaptation for effective video summarization. *Journal of Visual Communication & Image Representation*, 21:930–938.

[Rodríguez y otros, 2014] Rodríguez, I., Velázquez, C. O. R., y de la Cruz, A. V. (2014). Clusterización de alta disponibilidad y balanceo de carga en bases de datos de contenidos audiovisuales. En *Conferencia Científica UCIENCIA*.

[Rubner y otros, 2001] Rubner, Y., Puzicha, J., Tomasi, C., y Buhmann, J. M. (2001). Empirical evaluation of dissimilarity measures for color and texture. *Computer vision and image understanding*, 84(1):25–43.

[Santini, 2007] Santini, S. (2007). Who needs video summarization anyway? En *International Conference on Semantic Computing*. IEEE Comptuter Society.

[Sasonkgo, 2011] Sasonkgo, J. (2011). *Automatic Generation of Effective Video Summaries*. Tesis doctoral, Queensland University of Technology.

[Sáez Peña, 2006] Sáez Peña, E. (2006). *Segmentación automática de video.* Tesis doctoral, Universidad de Málaga, Málaga.

[Smeaton y otros, 2010] Smeaton, A. F., Over, P., y Doherty, A. R. (2010). Video shot boundary detection: Seven years of trecvid activity. *Computer Vision and Image Understanding*, 114(4):411–418.

[Song y otros, 2014] Song, G.-H., Ji, Q.-G., Lu, Z.-M., Fang, Z.-D., y Xie, Z.-H. (2014). A novel video abstraction method based on fast clustering of the regions of interest in key frames. *AEU-International Journal of Electronics and Communications.*

[Soriano, 1998] Soriano, R. M. (1998). *Introducción al Procesamiento y Análisis Digital de Imágenes.* Departamento de Ciencias de la COmputación e Inteligencia Artificial. Universidad de Granada.

[1995] Stroustrup, Bjarne and others (1995). *The C++ programming language.* Pearson Education India.

[Su, 1994] Su, L. T. (1994). The relevance of recall and precision in user evaluation. *Journal of the American Society for Information Science*, 45(3):207–217.

[Suárez Pérez y otros, 2012] Suárez Pérez, J. M., Fuentes Díaz, A., y Becerra Torreira, Y. (2012). Arquitectura para sistema gestor de procesos de media. *Revista Digital Sociedad de la Información*, 38:15.

[Swain y Ballard, 1991] Swain, M. J. y Ballard, D. H. (1991). Color indexing. *International journal of computer vision*, 7(1):11–32.

[Thounaojam y otros, 2014] Thounaojam, D., Trivedi, A., Manglem Singh, K., y Roy, S. (2014). A survey on video segmentation. En Mohapatra, D. P. y Patnaik, S., editores, *Advances in Intelligent Systems and Computing*, volumen 243, pp. 903–912. Springer India.

[Truong y Venkatesh, 2007] Truong, B. T. y Venkatesh, S. (2007). Video abstraction: A systematic review and classification. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 3(1):1–37.

[Tuytelaars y Mikolajczyk, 2008] Tuytelaars, T. y Mikolajczyk, K. (2008). Local invariant feature detectors: a survey. *Foundations and Trends in Computer Graphics and Vision*, 3(3):177–280.

[Valdés y Martínez, 2008] Valdés, V. y Martínez, J. M. (2008). On video abstraction system´s architectures and modelling. En *Semantic Multimedia*, pp. 164–177. Springer.

[Veeraraghavan y otros, 2005] Veeraraghavan, A., Roy-Chowdhury, A. K., y Chellappa, R. (2005). Matching shape sequences in video with applications in human movement analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(12):1896–1909.

[Wan y Qin, 2010] Wan, T. y Qin, Z. (2010). A new technique for summarizing video sequences through histogram evolution. En *Signal Processing and Communications (SPCOM), 2010 International Conference on*, pp. 1–5. IEEE.

[Weiming y otros, 2011] Weiming, H., Nianhua, X., Li, L., Xianglin, Z., y Maybank, S. (2011). A survey on visual content-based video indexing and retrieval. En *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, volumen 41, pp. 797–819.

[Witten y otros, 2011] Witten, I. H., Frank, E., y Hall, M. A. (2011). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, third edición.

[Xiang y otros, 2011] Xiang, J., Junwei, H., Xintao, H., Kaiming, L., Fan, D., Jun, F., Lei, G., y Tianming, L. (2011). Retrieving video shots in semantic brain imaging space using manifold-ranking. En *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pp. 3633–3636.

[Xiang-Wei y otros, 2009] Xiang-Wei, L., Ming-Xin, Z., Shuang-Ping, Z., y Ya-Ling, Z. (2009). A novel dynamic video sumarization approach based on rough sets in comprensed domain. *International Technology Journal*, 8(3):6–.

[Xu y C. Wunsh, 2009] Xu, R. y C. Wunsh, D. (2009). *Clustering*. Wiley.

[Yager, 1988] Yager, R. R. (1988). On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *Systems, Man and Cybernetics, IEEE Transactions on*, 18(1):183–190.

[Yinzi, 2010] Yinzi, C. (2010). A temporal video segmentation and summary generation method based on shots abrupt and gradual transition boundary detecting. En *Second International Conference on Communication Software and Network*, pp. –.

[Zadeh, 1965] Zadeh, L. A. (1965). Fuzzy sets. *Information and control*, 8(3):338–353.

[Zadeh, 1975] Zadeh, L. A. (1975). Fuzzy logic and approximate reasoning. *Synthese*, 30(3-4):407–428.

[Zhang y Wang, 2012] Zhang, C. y Wang, W. (2012). A robust and efficient shot boundary detection approach based on fisher criterion. pp. 701–704–.

[Zhe-Ming y Yong, 2013] Zhe-Ming, L. y Yong, S. (2013). Fast video shot boundary detection based on svd and pattern matching. *Image Processing, IEEE Transactions on*, 22(12):5136–5145–.

[Zhu y otros, 2003] Zhu, X., Fan, J., Elmagarmid, A. K., y Wu, X. (2003). Hierarchical video content description and summarization using unified semantic and visual similarity. *Multimedia Systems*, 9(1):31–53.

[Zhu y otros, 2004] Zhu, X., Wu, X., Fan, J., K. Elmagarmid, A., y Aref, W. G. (2004). Exploring video content structure for hierarchical summarization. *Multimedia Systems*, 10:98–115.

Acrónimos y siglas

AGORAV Plataforma de Gestión, Catalogación y Publicación Web de Contenidos Audiovisuales.

API Interfaz de Programación de Aplicación, del inglés, *Application Programming Interface*. .

FP Falsos Positivos, del inglés, *False Positive*.

FN Falsos Negativos, del inglés, *False Negative*.

GEYSED Centro de Geoinformática y Señales Digitales.

GOP Grupo de Imagénes, del inglés, *Group of Pictures*.

HSV Matiz, Saturación y Valor de intensidad, del inglés, *Hue, Saturation and Intesity Value*.

IDE Entorno de Desarrollo Integrado, del inglés, *Integrated Development Environment*.

MPEG Grupo Experto de Imágenes en Movimiento, del inglés, *Moving Picture Experts Group*.

PRIMICIA Plataforma de Televisión Informativa.

RAM Memoria de Acceso Aleatorio, del inglés, *Random Access Memory*.

RGB Rojo, Verde y Azul, del inglés, *Red, Green and Blue*.

SIAV Sistema de Gestión, Procesamiento y Transmisión de Contenidos Audiovisuales.

STCV Sistema de Transmisión de Canales Virtuales. .

TP Verdaderos Positivos, del inglés, *True Positive*.

TN Verdaderos Negativos, del inglés, *True Negative*.

UCI Universidad de las Ciencias Informáticas.

.

YIQ Información en escala de grises, Matiz y Saturación

Anexos

El anexos a continuación brinda un conjunto de elementos como apoyo y complemento de la investigación realizada.

Encuesta aplicada para la validación de escalabilidad y usabilidad

Estimado compañero, la siguiente encuesta es anónima, se aplica como instrumento de diagnóstico. Por favor le solicitamos que emita su valoración con la mayor sinceridad posible y le agradecemos de antemano su colaboración.

En cada pregunta usted posee una tabla en la que debe asignar un número entero entre 0 y 10 a cada uno de los valores (v_1, v_2, v_3), siendo 0 y 10 los valores de menor y mayor peso respectivamente. Debe asignar el valor central (v_2) estableciendo su criterio real para cada pregunta y los valores (v_1, v_3) deben ser asignados como los criterios mínimos (v_1) y máximos (v_3) que usted asignaría según la pregunta. Debe cumplirse que $(v_1 \leq v_2 \leq v_3)$. Si posee alguna duda por favor pregunte antes de responder.

Pregunta 1: ¿ Considera que el resultado muestra una síntesis del contenido de la secuencia de video que ha decidido procesar?

v_1	v_2	v_3

Pregunta 2: Si usted necesita realizar una descripción del video o tomar una decisión sobre el mismo, utilizando los resúmenes resultantes. ¿ En qué nivel le resultarían útiles los resúmenes?

v_1	v_2	v_3

Pregunta 3: ¿ Cómo valora la utilidad de generar varios resúmenes de la misma secuencia?

v_1	v_2	v_3

Pregunta 4: ¿ Considera que los resúmenes generados son atractivos para el usuario?

v_1	v_2	v_3