

Universidad de las Ciencias Informáticas

Facultad 5



“Módulo de adquisición de video para herramienta de autoría de contenidos de realidad aumentada”

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERÍA EN CIENCIAS INFORMÁTICAS

Autor:

Alejandro Coba Espinosa.

Tutores:

Ing. Mileydi Moreno Mirabal.

Ing. Ernesto de la Cruz Guevara Ramírez.

Ciudad de la Habana

2011

DECLARACIÓN DE AUTORÍA

DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Alejandro Coba Espinosa

Firma del Autor

Ing. Mileydi Moreno Mirabal

Firma del Tutor

Ing. Ernesto de la Cruz Guevara Ramírez

Firma del Tutor

DATOS DE CONTACTO

DATOS DE CONTACTO

Nombre y Apellidos: Mileydi Moreno Mirabal

Edad: 25 años

Ciudadanía: cubano

Institución: Universidad de las Ciencias Informáticas (UCI)

Título: Ingeniero en Ciencias Informáticas

Categoría Docente: Profesor Instructor

E-mail: mmirabal@uci.cu

Graduado en Ingeniería en Ciencias Informáticas, en la Universidad de las Ciencias Informáticas, con cuatro años de experiencia en el tema de la Gráfica Computacional. Actualmente Profesor Instructor de la Universidad de las Ciencias Informáticas impartiendo la asignatura de Geometría computacional.

Nombre y Apellidos: Ernesto de la Cruz Guevara Ramírez.

Edad: 26 años

Ciudadanía: cubano

Institución: Universidad de las Ciencias Informáticas (UCI)

Título: Ingeniero en Ciencias Informáticas

Categoría Docente: Profesor Instructor

E-mail: elguevara@uci.cu

Graduado en Ingeniería en Ciencias Informáticas, en la Universidad de las Ciencias Informáticas, con cuatro años de experiencia en el tema de la Gráfica Computacional. Actualmente Profesor Instructor de la Universidad de las Ciencias Informáticas impartiendo la asignatura de Gráficos por Computadoras y Geometría computacional.

AGRADECIMIENTOS

AGRADECIMIENTOS

Primero que todo agradecer a mi mamá que ha sabido ser mi madre y mi padre desde mi infancia, por hacer de mi lo que hasta hoy he sido y seré, a ti mamá muchas gracias.

A mi papá que a pesar de no estar físicamente presente, siempre ha sido un ejemplo a seguir por mí.

A mi abuela Blanca y mi tío Pablito por todo su afecto y por estar siempre al pendiente de mí, así como de mis estudios en todo momento, a ustedes les debo gran parte de lo que soy y seré en la vida.

A mi familia en general.

A mis amigos del barrio Eduardo, Niamey, Vladimir, Ale, Patato y El Chino que me han brindado su amistad desde la infancia.

A mis amigos de la universidad especialmente Enrique, José Antonio, el Charly, el Consu, Adrian, Mateo, Yuniel, Tony y el Ley con los que he tenido el placer de compartir en estos últimos años.

A todos aquellos que en algún momento de mi vida me han ayudado o han compartido parte de su tiempo conmigo.

Sin todos ellos habría sido imposible llegar hasta aquí.

Muchas Gracias.

DEDICATORIA

DEDICATORIA

A mi mamá y a toda mi familia y amigos.

A mi papá aunque ya no esté.

Alejandro.

Resumen

Resumen

Este trabajo propone una biblioteca para la adquisición de videos digitales. Surge a partir de la necesidad de realizar un componente capaz de obtener información de la realidad, que luego será integrado a un Sistema de Realidad Aumentada.

El trabajo se centra en los módulos para la adquisición de videos digitales desde diversas localizaciones en Sistemas de Realidad Aumentada. Se propone como variante de solución el empleo de la biblioteca OpenCv como biblioteca de desarrollo, para leer y reproducir los ficheros de videos digitales desde diversas localizaciones.

El presente trabajo se propone obtener una biblioteca que brinda las funcionalidades básicas para la manipulación de videos digitales, que pueda ser utilizada por todo aquel que pretenda incorporar video a un Sistema de Realidad Aumentada.

PALABRAS CLAVE

Video

Video Digital

Adquisición de Videos

OpenCv

Realidad Aumentada

Sistemas de Realidad Aumentada.

Summary

Summary

This paper proposes a library for the purchase of digital videos. Arises from the need for a component capable of obtaining information about reality, which is then integrated into a Augmented Reality System.

The work focuses on the forms for the purchase of digital videos from various locations in Augmented Reality Systems. It is proposed as an alternative solution the use of the OpenCv library and development library to read and play digital video files from various locations.

The present work aims at obtaining a library that provides basic functionality for handling digital video, which can be used by anyone who wishes to include video to Augmented Reality System.

Índice

Índice

Introducción	1
Capítulo 1 Fundamentación Teórica	4
1.1 Sistemas de Realidad Aumentada.	4
1.2 Videos digitales.	6
1.3 Formatos de video digital.	7
1.4 Formato AVI.	8
1.5 Módulos de adquisición de video.	9
1.6 Dispositivos de captura video digital.	10
1.7 Retos actuales de los módulos de adquisición de video.	14
1.8 Bibliotecas más utilizadas en el mundo para el trabajo con Realidad Aumentada.	15
1.8.1 OpenCv.	15
1.8.2 ARToolKit.....	16
1.8.3 EZFlar.....	17
1.8.4 FLARManager.	17
1.9 Descripción de la metodología de software utilizada	19
1.10 Fundamentación de los lenguajes y herramientas utilizadas.....	19
Capítulo 2 Solución Propuesta.....	23
2.1 Soluciones técnicas.....	23
2.2 Reglas del negocio.	24
2.3 Modelo de Dominio.....	25
2.3.1 Glosarios de Términos del Dominio.	25
2.4 Estructura principal de módulo.	26

Índice

2.5 Localización, captura, y reproducción del video.	28
2.6 Operaciones sobre el video.	31
2.6.1 Obtención de las características del video.	31
2.6.2 Funcionalidades a realizar sobre el video.	32
2.7 Utilidad.	33
2.8 Captura de requisitos.	34
2.8.1 Requisitos funcionales.	34
2.9 Diagrama de casos de uso del sistema.	36
2.9.1 Definición del actor del sistema.	36
2.9.2 Expansión de los casos de uso.	36
Capítulo 3: Diseño e implementación.	41
3.1 Diagrama de clases de diseño.	42
3.2 Descripción de las clases.	43
3.3 Diagramas de Secuencia.	46
3.4 Diagrama de componentes.	49
3.5 Diagrama de despliegue.	50
3.6 Discusión de los resultados.	51
Conclusiones	53
Recomendaciones	54
Referencias Bibliográficas	55
Anexos	58
Glosario de Términos	61

Índice de figuras

Índice de figuras

<i>Figura 1: Subsistemas fundamentales de un Sistema de Realidad Aumentada.</i>	5
<i>Figura 2: Imágenes de dispositivos de visualización.</i>	5
<i>Figura 3: Imagen de cámara web.</i>	14
<i>Figura 4: Ejemplo de trabajo con ARToolKit</i>	17
<i>Figura 5: Ejemplo de trabajo con FLARManager.</i>	18
<i>Figura 6: Modelo de Dominio.</i>	25
<i>Figura 7: Estructura principal del módulo de adquisición de video.</i>	27
<i>Figura 8: Diagrama de casos de uso.</i>	36
<i>Figura 9: Diagrama de clases de diseño.</i>	42
<i>Figura 10: Diagrama de secuencia del diseño. Cargar video desde distintas localizaciones.</i>	46
<i>Figura 11: Diagrama de secuencia del diseño. Reproducir video.</i>	47
<i>Figura 12: Diagrama de secuencia del diseño. Calcular características del(los) video(s).</i>	47
<i>Figura 13: Diagrama de secuencia del diseño. Realizar acciones sobre el(los) video(s).</i>	48
<i>Figura 14: Diagrama de componentes.</i>	50
<i>Figura 15: Diagrama de despliegue.</i>	51
<i>Figura 16: Adquisición de varios videos de archivo, red y cámara.</i>	58
<i>Figura 17: Adquisición de video desde dos cámaras simultáneamente.</i>	58
<i>Figura 18: Cálculo de las características de un video.</i>	59
<i>Figura 19: Realización de funciones sobre un video.</i>	59
<i>Figura 20: Adquisición de video desde un archivo</i>	60

Índice de tablas

Índice de tablas

<i>Tabla 1: Distintos tipos de dominios para cámaras.....</i>	<i>30</i>
<i>Tabla 2: Definición de los actores.....</i>	<i>36</i>
<i>Tabla 3: Expansión del Caso de Uso (Cargar video desde distintas localizaciones).</i>	<i>38</i>
<i>Tabla 4: Expansión del Caso de Uso (Reproducir video(s)).....</i>	<i>38</i>
<i>Tabla 5: Expansión del Caso de Uso (Calcular características del(los) video(s)).</i>	<i>39</i>
<i>Tabla 6: Expansión del Caso de Uso (Realizar funcionalidades sobre el(los) video(s)).</i>	<i>40</i>
<i>Tabla 7: Descripción de la clase (cRender).....</i>	<i>45</i>
<i>Tabla 8: Descripción de la clase (Ventana).....</i>	<i>45</i>
<i>Tabla 9: Velocidad de reproducción del modulo para uno y mas videos.....</i>	<i>51</i>

Introducción

A pesar de haber surgido hace ya algunos años, el fenómeno denominado Realidad Aumentada (RA) continúa revolucionando el mundo, no sólo de la informática sino también de diversas áreas como la medicina, la arquitectura, la educación, etc.

En la actualidad la aplicación de Sistemas de Realidad Aumentada (SRA) se ha extendido a una amplia gama de sectores. Un factor primordial en su flujo de trabajo es la captura de la información que luego es analizada en busca de pistas para insertar la información virtual. Esta captura de la información se logra mediante la utilización de módulos de adquisición de video.

La Universidad de las Ciencias Informáticas (UCI) es una de las instituciones a la cabeza en el desarrollo de la informática en Cuba. El Centro de Informática Industrial (CEDIN) de esta universidad cuenta con la Línea de núcleo gráfico del departamento de visualización y realidad virtual, el cual se encarga de desarrollar una Sistema de Realidad Aumentada. Este sistema actualmente trabaja con un módulo de adquisición de video llamado Direct Show Video Library (DSVL), que ofrece varios inconvenientes, ya que no permite cargar videos ubicados en archivos persistentes o recursos compartidos de red, ni reproducir varias cámaras simultáneamente.

Analizando detenidamente la problemática anterior, este trabajo propone como **problema científico** a resolver:

¿Cómo lograr una buena adquisición de flujo de video utilizando varios tipos de fuentes (Archivos, dispositivo de captura y video en red)?

Como **objeto de estudio**: Adquisición de información del mundo real para Realidad Aumentada.

Con el fin de resolver el problema científico se trazó el siguiente **objetivo general**:

Introducción

Desarrollar un complemento para la herramienta de autoría de contenidos de realidad aumentada, mediante el cual se van a obtener los videos para la realización del proceso de reconocimiento de patrones.

Y como **campo de acción:** Sistemas de adquisición de video para Realidad Aumentada.

Para el total cumplimiento del objetivo general se trazan las siguientes tareas investigativas:

1. Análisis de la bibliografía existente y estudio del estado del arte sobre desarrollo de componentes para Realidad Aumentada, a fin de conocer en qué estado se encuentra el mismo y cuáles son sus principales tendencias.
2. Caracterización de las diferentes formas de adquisición de información para Realidad Aumentada.
3. Seleccionar biblioteca de adquisición de video para dar solución al problema planteado en la investigación.
4. Construcción de un componente con interfaces bien definidas para la adquisición de video (Archivo, dispositivo, recurso compartido de red).
5. Implementación de una aplicación demostrativa que incluya las funcionalidades que se implementarán en el componente.

Métodos Teóricos:

Histórico-lógico: Este método permitió realizar la primera parte de la investigación concerniente al análisis bibliográfico del tema, lo cual posibilitó evaluar la bibliografía para determinar los conceptos necesarios de la Realidad Aumentada, así como la adquisición de video en SRA. De esta forma se obtuvo un conocimiento del estado actual en que se encuentran las mismas.

Analítico – Sintético: Facilitó su estudio mediante la descomposición en sus principales partes y seguidamente resumió los distintos enfoques que se le dieron al desarrollo de componentes para Sistemas de Realidad Aumentada.

Introducción

Inductivo – Deductivo: Comenzó con el estudio de los procesos de desarrollo de distintos componentes específicos para Realidad Aumentada que fueron analizados con anterioridad, para obtener las ideas generales que permitan determinar las acciones a realizar para implementar la aplicación.

El presente trabajo de diploma está estructurado de la siguiente forma: Resumen, Introducción, tres capítulos de contenido, Conclusiones, Recomendaciones, Referencias bibliográficas, Bibliografía consultada, Glosario de abreviaturas, Glosario de términos y los Apéndices. A continuación se hace una breve descripción del contenido de cada uno de los capítulos:

Capítulo 1 **Fundamentación Teórica:** Se hace un breve análisis de los Sistemas de Realidad Aumentada. Se estudian los formatos de video digital enfatizando en el formato AVI. También se analizan los dispositivos de adquisición videos digitales, así como los módulos de adquisición de video y las bibliotecas y herramientas más utilizadas en el desarrollo de aplicaciones de Realidad Aumentada.

Capítulo 2 **Solución Propuesta:** En este capítulo se proponen las soluciones técnicas para la creación de funcionalidades concernientes a la adquisición de información en Realidad Aumentada. Además se define la estructura principal del prototipo funcional elaborado y se explica de forma detallada la solución. Se hace una conceptualización a través del modelo de dominio, así como una explicación detallada de este para su mejor entendimiento. También se especifican los requisitos funcionales y no funcionales, y por último se lleva a cabo la expansión de los casos de uso.

Capítulo 3 **Diseño e implementación:** En este capítulo se tratan los aspectos fundamentales del diseño e implementación de la solución. Se muestran los diagramas de clases de diseño, los diagramas de secuencia, las descripciones de las clases de diseño, así como el diagrama de componentes y de despliegue de la solución. Por último se hace una discusión de los resultados obtenidos.

Capítulo 1 Fundamentación Teórica

En este capítulo se hace un estudio acerca de los videos digitales así como de los formatos de videos digitales más conocidos, haciendo énfasis en el formato AVI. También se caracterizan algunos de los dispositivos más conocidos de adquisición de videos digitales. El autor presenta las características de las herramientas de autoría de contenidos de RA que considera más importantes, resaltando la adquisición de video como uno de sus flujos principales. Se hace un estudio de las principales bibliotecas utilizadas en el mundo para el trabajo con Realidad Aumentada. Además se abordan los retos fundamentales y las tendencias actuales de los Módulos de adquisición de video en el mundo, en Cuba y en la UCI, lo cual constituyó la base teórica necesaria para llevar a cabo la presente investigación.

1.1 Sistemas de Realidad Aumentada.

Según (4) todo Sistema de Realidad Aumentada (SRA) debe cumplir con tres características al mismo tiempo:

- Combinan lo real y lo virtual;
- Son interactivas en tiempo real;
- Llevan a cabo la inscripción en el espacio 3D.

Capítulo 1 Fundamentación Teórica

En cuanto a su funcionamiento, los SRA tienen tres subsistemas fundamentales:
(5)

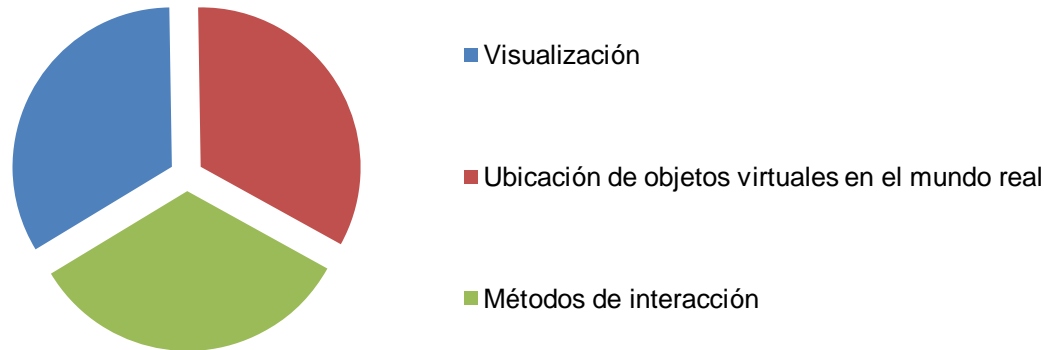


Figura 1: Subsistemas fundamentales de un Sistema de Realidad Aumentada.

Visualización: Se logra con el uso de dispositivos de visualización similares a los de RV. Algunos de estos dispositivos son cascos, gafas y monitores. Los cascos y las gafas se componen por pantallas de cristal líquido funcionando como si fueran lentes transparentes para que pueda observarse el mundo con la adición de los objetos virtuales. (5)



Figura 2: Imágenes de dispositivos de visualización.

Ubicación de objetos virtuales en el mundo real: Consiste en lograr que los objetos virtuales puedan registrarse con el mundo real, de tal forma que cuando el usuario se mueva los objetos parezcan conservar su posición. (5)

Interacción: Consiste en métodos para manipular o modificar tales objetos. (5) El principal objetivo en el diseño de sistemas de RA es que estos lleguen a ser tan portables, ligeros, pequeños y robustos como sea posible y que permitan al

Capítulo 1 Fundamentación Teórica

usuario explorar cualquier entorno no preparado, ya sea interior o exterior, sin ninguna restricción.

Es necesario aclarar que dentro del subsistema de visualización se incluye la obtención de información del mundo real; posteriormente es analizada en busca de pistas para la inclusión de elementos virtuales. Esto se puede lograr mediante dispositivos de captura de video digital.

1.2 Videos digitales.

El video digital es un tipo de sistema de grabación de video que funciona usando una representación digital de la señal de vídeo. Los mismos pueden ser grabados en cintas magnéticas o directamente en discos duros y memorias flash. Estos se introdujeron por primera vez en 1983 con el formato D-1 de Sony. Aunque al principio su calidad era baja, mejoró rápidamente con la incorporación de estándares de reproducción como MPEG-1 y MPEG-2 y el formato de cinta DV , el cual permite grabar directamente datos digitales, ya que permite utilizar completamente sistemas de edición no lineal en ordenadores de sobremesa.

Otra forma de obtener un video digital es realizando el proceso de digitalización a un video analógico. Para digitalizar una señal de video analógico es necesario muestrear todas las líneas de video activo. La información de brillo y color son tratadas de forma diferente por el sistema visual humano, ya que es más sensible al brillo que al color. Con lo que se usa un componente especial para representar la información del brillo, la luminancia, una para el color y la saturación, la crominancia. Cada muestra de color se codifica en señal Y-U-V (Y- luminancia, U y V crominancia) partiendo de los valores del sistema RGB. Con este sistema las diferencias de color pueden ser muestreadas sin resultados visibles, lo que permite que la misma información sea codificada con menos ancho de banda.

Las imágenes de video están compuestas de información en el dominio del espacio y el tiempo. La información en el dominio del espacio es provista por los pixeles, y la información en el dominio del tiempo es provista por imágenes que

Capítulo 1 Fundamentación Teórica

cambian en el tiempo. Puesto que los cambios entre cuadros colindantes son diminutos, los objetos aparentan moverse suavemente.

El valor de luminancia de cada pixel es cuantificado con ocho bits para el caso de imágenes blanco y negro. En el caso de imágenes de color, cada pixel mantiene la información de color asociada; una imagen completa es una composición de tres fotogramas, uno para cada componente de color, así los tres elementos de la información de luminancia son cuantificados a ocho bits. (1)

El vídeo digital se usa también fuera de la creación de películas, en teléfonos móviles modernos, sistemas de videoconferencia y en Sistemas de Realidad Virtual y Aumentada. Además se usa para la distribución de videos en Internet, incluyendo el vídeo streaming y la distribución de películas entre iguales.

1.3 Formatos de video digital.

La reproducción de video es ya un elemento común en los ordenadores modernos, siendo esta una de las tareas que más recursos consume. La sensación de movimiento se consigue con secuencias de imágenes con una velocidad de unos 30 fotogramas por segundo. Si el objetivo es video a través de Internet, las limitaciones del ancho de banda de la red presentan muchos problemas. Las posibilidades se reducen de nuevo al empleo de sistemas de compresión y a la reducción del tamaño de las ventanas de video y del número de fotogramas por segundo.

Una fotografía sin comprimir de 800x600 píxeles de resolución ocupa aproximadamente 1,3 Mb. Así pues, secuencias de video de este tamaño y con 30 fotogramas por segundo generarían ficheros de video con un tamaño de 390 Mb para 10 segundos ó 11,4 Gigabytes para 5 minutos. Por tanto, son imprescindibles drásticos sistemas de compresión para el manejo de video. (2)

Los estándares de video digital más conocidos son: MPEG, QuickTime, MOV, AVI, real video, ASF, entre otros.

Capítulo 1 Fundamentación Teórica

1.4 Formato AVI.

AVI es el acrónimo de Audio Video Interleave (intercalado de audio y video). Se trata de un formato de archivo que actúa como contenedor de flujos de datos de audio y video.

Intercalado significa que en un fichero AVI los datos de audio y video son almacenados consecutivamente en capas (un segmento de datos de video es seguido inmediatamente por otro de audio). Es el formato más extendido para el manejo de datos de audio/video en una PC. (3)

El formato avi permite almacenar simultáneamente un flujo de datos de video y varios flujos de audio. El formato concreto de estos flujos no es objeto del formato AVI y es interpretado por un programa externo denominado códec. Es decir, el audio y el video contenidos en el AVI pueden estar en cualquier formato. Por eso se le considera un formato contenedor.

Para trabajar con archivos de video de extensión (.avi), generalmente se abre el archivo directamente, no se emplea ninguna técnica para codificar el archivo de video o cambiar su extensión con el objetivo de proteger los derechos del autor, aunque es posible realizar estas operaciones.

Por lo general un fichero (.avi) guarda solamente información de audio y de video, se utilizan arreglos para almacenar esta información. Lo importante es especificar qué tipo de información se cargará, si pertenece a video o audio. Una vez que se conoce el tamaño del arreglo es posible hacer ciclos.

Para facilitar abrir y reproducir los ficheros AVI, se pueden utilizar funciones ya existentes, por lo que la información es obtenida simplemente por el empleo de una función y no es necesario manejar la descompresión del video.

La información del video que se maneja, es principalmente datos como las dimensiones del AVI (ancho y altura), el número del último fotograma, así como cuántos milisegundos se muestran cada fotograma para mantener la velocidad.

Capítulo 1 Fundamentación Teórica

Cuando el sistema sea inicializado, podemos hacer un ciclo que arranca y muestra cada fotograma de video, estos se guardan en un buffer y se convertirán cada uno en una imagen.

Solamente hay que especificar el formato en el que se almacena inicialmente y el formato en el que se convertirán los fotogramas de la animación dependiendo del sistema que se empleará para pintar.

Se organizan los fotogramas de manera que en la memoria solo se necesita uno a la vez, permitiendo la reproducción de archivos grandes.

1.5 Módulos de adquisición de video.

Los módulos de adquisición de video son casi imprescindibles en la confección de sistemas tanto de Realidad Virtual como de Realidad Aumentada y en general de todo sistema que utilice visión por computadoras.

Dos de los módulos de adquisición de video más utilizados actualmente en el mundo son DSVL (Direct Show Video Library por sus siglas en inglés) y FFmpeg.

Direct Show Video Library (DSVL) es una biblioteca que se utiliza para la adquisición de video en Sistema de Realidad Aumentada. La misma plantea que mediante su utilización se puede lograr la adquisición de video desde varias cámaras simultáneamente, lo cual en la práctica no se ha podido lograr ya que solo es posible cargar video desde una cámara. Esta biblioteca no brinda la posibilidad de adquirir videos desde archivos persistentes en discos duros ni en recursos compartidos de red, tampoco de realizar acciones sobre los videos ni calcular características referentes a los mismos. Además DSVL solo se puede utilizar en Windows.

FFmpeg plantea que con su utilización se puede lograr la adquisición de audio y video desde diversas fuentes. Posibilitando usar como entrada ficheros, flujos, entrada de tarjeta de TV, cámara web, etc. A pesar de posibilitar entrada desde

Capítulo 1 Fundamentación Teórica

más fuentes que DSVL, tampoco logra la adquisición simultánea de varios videos, ya sea desde cámaras, archivos o recursos de red. FFMPEG posibilita la integración con OpenCv ya que esta biblioteca trabaja por debajo con ffmpeg para poder leer los vídeos o trabajar con cámaras web en algunos casos cuando la dirección es por string.

1.6 Dispositivos de captura video digital.

Los dispositivos de captura de video digital son muy utilizados en la confección de Sistemas de Realidad Aumentada ya que la captura de información real constituye un aspecto primordial en su flujo de trabajo. Existen diversos tipos de cámaras de captura de videos digitales, algunas muy avanzadas; pero esto también influye en su costo. A continuación conoceremos más de cerca algunos de estos dispositivos.

Cámaras digitales

Una cámara digital es una cámara que, en vez de capturar y almacenar fotogramas en películas fotográficas como las cámaras convencionales, lo hace digitalmente mediante un dispositivo electrónico o en cinta magnética usando un formato analógico como muchas cámaras de video.

Existe gran variedad de cámaras digitales, las que varían desde formas y tamaños hasta en su funcionamiento.

Una variedad de cámaras digitales son las cámaras IP, las cuales se utilizan mucho en aspectos relacionados con la vigilancia.

Las cámaras IP se conectan directamente a una conexión LAN (RJ45) de su instalación de internet, red doméstica, inalámbrica o wifi y llevan incorporado un servidor Web. Son cámaras totalmente autónomas del ordenador, se les asigna una dirección IP interna, y tan sencillo como teclear esa dirección IP desde cualquier navegador para acceder a la cámara y disponer de los menús que permiten todo tipo de funciones; visionar, realizar grabaciones, escuchar,

Capítulo 1 Fundamentación Teórica

alarmas, etc. Vd. puede controlar su hogar, oficina o negocio desde cualquier lugar que disponga de internet.

Una cámara de red incorpora su propio miniordenador, lo que le permite emitir vídeo por sí misma. Además de comprimir el vídeo y enviarlo, puede tener una gran variedad de funciones:

- Envío de correos electrónicos con imágenes.
- Activación mediante movimiento de la imagen.
- Activación mediante movimiento de sólo una parte de la imagen.
- Creación una máscara en la imagen, para ocultar parte de ella o colocar un logo. O simplemente por adornar.
- Activación a través de otros sensores.
- Control remoto para mover la cámara y apuntar a una zona.
- Programación de una secuencia de movimientos en la propia cámara.
- Posibilidad de guardar y emitir los momentos anteriores a un evento.
- Utilización de diferente cantidad de fotogramas según la importancia de la secuencia. Para conservar ancho de banda.
- Actualización de las funciones por software.

Las cámaras IP permiten ver en tiempo real qué está pasando en un lugar, aunque esté a miles de kilómetros de distancia. Son cámaras de vídeo de gran calidad que tienen incluido un ordenador a través del que se conectan directamente a Internet.

Una cámara IP (o una cámara de red) es un dispositivo que contiene:

- Una cámara de vídeo de gran calidad, que capta las imágenes
- Un chip de compresión que prepara las imágenes para ser transmitidas por Internet, y
- Un ordenador que se conecta por sí mismo a Internet

Capítulo 1 Fundamentación Teórica

Las cámaras IP incorporan todas las funciones de una cámara de vídeo y añaden más prestaciones.

- La lente de la cámara enfoca la imagen en el sensor de imagen (CCD). Antes de llegar al sensor, la imagen pasa por el filtro óptico que elimina cualquier luz infrarroja y muestra los colores correctos.
- Actualmente están apareciendo cámaras día/noche que disponen de un filtro de infrarrojos automático, este filtro se coloca delante del CCD sólo cuando las condiciones de luz son adecuadas proporcionándonos de esta manera imágenes en color, cuando las condiciones de luz bajan este filtro se desplaza y la cámara emite la señal en blanco y negro produciendo más luminosidad y de esta manera podemos iluminar la escena con luz infrarroja y ver en total oscuridad.
- El sensor de imagen convierte la imagen, que está compuesta por información lumínica, en señales eléctricas. Estas señales eléctricas se encuentran ya en un formato que puede ser comprimido y transferido a través de redes.
- Como las cámaras de vídeo convencionales, las cámaras IP gestionan la exposición (el nivel de luz de la imagen), el equilibrio de blancos (el ajuste de los niveles de color), la nitidez de la imagen y otros aspectos de la calidad de la imagen. Estas funciones las lleva a cabo el controlador de cámara y el chip de compresión de vídeo.
- Las cámaras IP comprimen la imagen digital en una imagen que contiene menos datos para permitir una transferencia más eficiente a través de la Red, cámaras MPEG4.

Dentro de las cámaras digitales se encuentran también las cámaras web (en inglés webcam), las que son utilizadas en muchos proyectos relacionados con la computación.

Una cámara web es una pequeña cámara digital que conectada a una computadora, puede capturar imágenes y transmitir las en la misma computadora o a través de Internet, esto puede ser a una página web o a otra u otras

Capítulo 1 Fundamentación Teórica

computadoras de forma privada. Estas cámaras necesitan de una computadora para poder transmitir las imágenes y normalmente se conectan a través de un puerto USB. Hay que tener en cuenta es que dicha cámara no tiene nada de especial, es como el resto de las cámaras digitales, lo que realmente le da el nombre de "cámara web" es el software que la acompaña.

El software de la cámara web toma un fotograma de la cámara cada cierto tiempo (puede ser una imagen estática cada medio segundo) y la envía a otro punto para ser visualizada. Si lo que se pretende es utilizar esas imágenes para construir un video, de calidad sin saltos de imagen, se necesitará que la cámara web alcance una tasa de unos 15 a 30 fotogramas por segundo. Las cámaras web normalmente están formadas por una lente, un sensor de imagen y la circuitería necesaria para manejarlos.

Existen distintos tipos de lentes, siendo las plásticas las más comunes. Los sensores de imagen pueden ser CCD (*charge coupled device*) o CMOS (*complementary metal oxide semiconductor*). Este último suele ser el habitual en cámaras de bajo coste, aunque eso no signifique necesariamente que cualquier cámara CCD sea mejor que alguna CMOS. Las cámaras web para usuarios medios suelen ofrecer una resolución VGA (640x480) con una tasa de unos 30 fotogramas por segundo, si bien en la actualidad están ofreciendo resoluciones medias de 1 a 1,3 MP, en estos momentos las cámaras de gama alta cuentan con 3, 5, 8 o hasta 10 mega píxeles y son de alta definición.

La circuitería electrónica es la encargada de leer la imagen del sensor y transmitirla a la computadora. Algunas cámaras usan un sensor CMOS integrado con la circuitería en un único chip de silicio para ahorrar espacio y costes. El modo en que funciona el sensor es equivalente al de una cámara digital normal. Las cámaras web también pueden captar sonido, con una calidad mucho menor a la normalmente alcanzada por otros tipos de cámaras. (6)



Figura 3: Imagen de cámara web.

1.7 Retos actuales de los módulos de adquisición de video.

Son muchos los retos que enfrentan hoy los módulos de adquisición de video (MAV) tanto en el mundo, en Cuba como en la UCI. Entre los retos más importantes que enfrentan hoy los MAV resaltan los siguientes:

- Lograr adquisición de video desde diversas fuentes (archivos, dispositivo de captura, red).
- Permitir la reproducción de videos desde varios dispositivos de captura simultáneamente.
- Lograr que la velocidad de reproducción de los videos no se vea afectada en dependencia de la cantidad de videos en reproducción.

En el mundo se han llevado a cabo numerosos procesos investigativos, lo que ha posibilitado el desarrollo de gran número de componentes para la adquisición de información utilizando sensores ópticos, pero hasta ahora ninguno se ajusta a los requerimientos de nuestra línea de desarrollo, como por ejemplo cargar videos desde distintas cámaras al mismo tiempo, así como desde la red o desde un archivo persistente, entre otras funcionalidades.

También en nuestra facultad se creó una biblioteca para la adquisición de video destinada a aplicaciones de Realidad Virtual (RV), cuya función es la

Capítulo 1 Fundamentación Teórica

manipulación de videos digitales. Esta biblioteca ofrece la posibilidad de cargar videos desde un archivo persistente en el disco duro, y realizar diversas acciones sobre este, lo cual cumple con una de las restricciones de nuestra línea de desarrollo; pero no posibilita cargar videos desde cámaras web así como desde recursos compartidos de red.

1.8 Bibliotecas más utilizadas en el mundo para el trabajo con Realidad Aumentada.

La Realidad Aumentada a pesar de ser una tecnología relativamente joven, ya cuenta con una variada gama de bibliotecas que han sido diseñadas para la creación de Sistemas de Realidad Aumentada. Existen bibliotecas para RA enfocadas a la tecnología móvil, así como para la RA en la web y por supuesto, para las aplicaciones de escritorio. En los siguientes sub epígrafes el autor caracteriza a grandes rasgos algunas de las bibliotecas más utilizadas en el mundo para la creación de Sistemas de Realidad Aumentada.

1.8.1 OpenCv.

OpenCV (Open Source Computer Vision Library) es una biblioteca de código abierto desarrollada por Intel. Proporciona un alto nivel funciones para el procesamiento de imágenes. Esta biblioteca permite a los programadores crear aplicaciones poderosas en el dominio de la visión por computadoras. OpenCV ofrece muchos tipos de datos de alto nivel como árboles, grafos, matrices, etc. Es multiplataforma. Con esta biblioteca podemos realizar operaciones básicas, análisis, análisis estructural, análisis de movimiento, reconocimiento de modelos 3D, reconstrucción 3D y calibración de cámaras, así como interfaces gráficas y de adquisición entre otras más. (7)

Esta biblioteca implementa una gran variedad de herramientas para la interpretación de la imagen. Es compatible con Intel Image Processing Library (IPL por sus siglas en inglés, no es más que un conjunto de bibliotecas de Intel

Capítulo 1 Fundamentación Teórica

que implementan diferentes funciones de procesamiento de imágenes. Está optimizada para procesadores de Intel, que implementa algunas operaciones en imágenes digitales (también funcionan con cualquier procesador compatible con Pentium). A pesar de primitivas como filtrado, estadísticas de la imagen, pirámides, OpenCV es principalmente una biblioteca que implementa algoritmos para las técnicas de la calibración (Calibración de la Cámara), detección de rasgos, para rastrear (Flujo Óptico), análisis de la forma (Geometría, Contorno que Procesa), análisis del movimiento (Plantillas del Movimiento, Estimadores), reconstrucción 3D (Transformación de vistas), segmentación de objetos y reconocimiento (Histograma, etc.). (8)

El rasgo esencial de la biblioteca junto con funcionalidad y la calidad es su desempeño. Los algoritmos están basados en estructuras de datos muy flexibles, acoplados con estructuras IPL. Más de la mitad de las funciones han sido optimizadas aprovechándose de la Arquitectura de Intel.

1.8.2 ARToolKit.

ARToolKit es una biblioteca que permite la creación de aplicaciones de realidad aumentada, en las que se superpone información virtual en el mundo real a través de dispositivos de captura y visualización. Para ello, utiliza las capacidades de seguimiento de vídeo, con el fin de calcular, en tiempo real, la posición de la cámara y la orientación relativa a la posición de los marcadores físicos. Una vez que la posición de la cámara real se sabe, la cámara virtual se puede colocar en el mismo punto y modelos 3d son sobrepuestos exactamente sobre el marcador real. Así ARToolKit resuelve dos de los principales problemas en la realidad aumentada, el seguimiento de punto de vista y la interacción objeto virtual.

ARToolKit incluye código de seguimiento que utiliza simples cuadros de color negro. Cuenta con la capacidad de utilizar cualquier patrón marcador cuadrado. Posee un código fácil de calibración de la cámara. Es bastante rápido para aplicaciones en tiempo real de RA. Es compatible con SGI IRIX, Linux, MacOS y

Capítulo 1 Fundamentación Teórica

distribuciones del sistema operativo Windows. ARToolKit fue distribuido con el código fuente completo. (9)



Figura 4: Ejemplo de trabajo con ARToolKit

1.8.3 EZFlar.

EZFlar es la biblioteca oficial que trabaja con Realidad Aumentada para crear el periódico brasileño "O Estado de S.Paulo". Posee un lenguaje Action Script de código abierto, diseñado para los desarrolladores interesados en crear aplicaciones de RA con alta velocidad. Aunque existen otras bibliotecas de código abierto y destinadas a la web como FLARManager, FlarToolkityPapervision3D. EZFlar va más allá y añade nuevas características, trayendo consigo un código limpio y buen rendimiento.(10)

1.8.4 FLARManager.

Capítulo 1 Fundamentación Teórica

FLARManager es una biblioteca ligera que hace que sea más fácil crear aplicaciones de realidad aumentada para Flash. Es compatible con una variedad de bibliotecas de seguimiento y frameworks3D (listados abajo), y proporciona un sistema más sólido basado en eventos para la gestión de marcas, actualización y eliminación. Se admite la detección y la gestión de varios patrones y marcadores múltiples de un patrón dado.

Bibliotecas compatibles que incluyen seguimiento:

- FLARToolkit
- flare*tracker
- flare*NFT

FLARManager aún no ofrece compatibilidad entre Bengala y Away3D/Lite, Sandy3D o Alternativa3D. Esta incompatibilidad imposibilita su utilización en este trabajo debido a que nuestra línea de desarrollo necesita interacción con elementos 3D.(11)



Figura 5: Ejemplo de trabajo con FLARManager.

Capítulo 1 Fundamentación Teórica

Una vez concluido el análisis de las bibliotecas anteriores se notó que tanto EZFlar como FLARManager son bibliotecas que están enfocadas al trabajo con flash. OpenCv y ARToolKit están relacionadas ya que este último incluye a OpenCv para realizar muchas de sus funciones, y OpenCv a su vez incluye varios codecs para lograr la obtención de videos desde varias fuentes.

1.9 Descripción de la metodología de software utilizada.

Para el desarrollo del prototipo funcional se usó la metodología del Proceso Unificado de Rational (RUP por sus siglas en inglés). Este proceso de ingeniería de software se basa en la modelación de sistemas informáticos usando la tecnología orientada a objetos, lo que provee un enfoque disciplinado para asignar tareas y responsabilidades dentro de una organización desarrolladora o cualquier proyecto de software.

Como lenguaje de modelado se seleccionó UML, lenguaje para especificar, construir y documentar los artefactos de un sistema con gran cantidad de software. UML proporciona una forma estándar de escribir los planos de un sistema, cubriendo tanto los detalles conceptuales tales como procesos del negocio y funciones del sistema; como los concretos tales como las clases escritas en un lenguaje de programación específico, esquemas de bases de datos y componentes de software reutilizable+(12)

Entre las principales ventajas de RUP se destacan:

1.10 Fundamentación de los lenguajes y herramientas utilizadas.

Para elaborar un programa informático sin importar de qué tipo, se hace necesario utilizar uno o varios lenguajes de programación así como herramientas de desarrollo para lograr las funcionalidades requeridas a través de estos. Tanto las herramientas como el lenguaje que se proponen en este epígrafe, serán puestos en marcha en todas las fases de desarrollo del módulo.

Capítulo 1 Fundamentación Teórica

A continuación se tratan brevemente el lenguaje y las herramientas necesarias para desarrollar el módulo de adquisición de video para un Sistema de Realidad Aumentada.

Lenguaje C++.

El lenguaje C++ es uno de los preferidos por la mayoría de los programadores para el trabajo con gráficos por computadora ya que posee diversas características que lo hacen mejor en este sentido. C++ es un lenguaje de programación orientado a objetos. Se suele decir que es un lenguaje híbrido, ya que permite la programación estructurada. Con la utilización de este lenguaje, el programador tiene el control total de lo que está haciendo, permitiendo una máxima eficiencia al no incorporar verificación de errores en tiempo de ejecución.

El código generado por los compiladores del lenguaje no debe incluir una sobrecarga de recursos computacionales, minimizando la implementación de recursos tales como el polimorfismo y la expansión de patrones. Los compiladores de C++ generan código nativo con un alto grado de optimización en memoria y velocidad, lo que lo convierte en uno de los lenguajes más eficientes.

Una de las características más interesantes del lenguaje es la sobrecarga de operadores. Esto significa que a los operadores intrínsecos del lenguaje se les puede redefinir la semántica: se pueden escribir funciones que en vez de tener un nombre, se asocian a un operador, que debe tener por lo menos un parámetro de tipo clase. Permite trabajar tanto a alto como bajo nivel lo que lo convierte en un lenguaje muy potente. Es uno de los lenguajes por excelencia para desarrollar programas de grafico por computadoras. (13)

Además de todas estas características que lo convierten en la mejor opción, se eligió este lenguaje de programación ya que una de las restricciones del proyecto plantea que el módulo debe estar hecho en C++.

Microsoft Visual Studio.NET 2008.

Capítulo 1 Fundamentación Teórica

Microsoft Visual Studio.NET 2008 está diseñado para aprovechar las ventajas que ofrece el sistema operativo "Windows Vista", posee mejoras de desempeño, escalabilidad y seguridad con respecto a la versión anterior, proporciona a los programadores un lenguaje orientado a objetos de probada eficacia para generar aplicaciones de alto rendimiento. Gracias a plantillas avanzadas, acceso a plataformas de bajo nivel y un compilador compatible con el estándar ANSI C++, que optimiza las compilaciones, ofrece funcionalidad para generar componentes sólidos, utilizando el lenguaje de programación C++. Soporta la biblioteca gráfica OpenGL que es utilizada por excelencia en la industria de los gráficos por computadoras y video-juegos. (14)

Visual Paradigm for UML 7.2 Enterprise Edition.

Visual Paradigm para UML (Lenguaje de Modelado Unificado) es una herramienta CASE profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones con gran calidad y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales, demostraciones interactivas y proyectos UML. Además soporta las últimas versiones de UML y la notación y modelado de procesos de negocios desde un grupo administrador de objetos.

A continuación se muestran algunas de sus características más importantes:

- Es multiplataforma.
- Soporta UML versión 2.1.
- Posibilita la realización de ingeniería inversa a Java, C++, Esquemas XML, entre otros.
- Cuenta con un editor de Detalles de los Casos de Uso, para la especificación de los detalles de los casos de uso, incluyendo la

Capítulo 1 Fundamentación Teórica

especificación del modelo general y de las descripciones de los casos de uso.

- Posee una distribución automática de diagramas, lo que se traduce en reorganización de las figuras y conectores de los diagramas UML.(15)

La RA es una nueva modalidad que poco a poco está siendo incorporada en muchas esferas de la vida. Mediante este capítulo se trataron algunos conceptos de RA. Además se analizaron algunas bibliotecas utilizadas para lograr RA, y se hizo un enfoque a los Sistemas de Realidad Aumentada, tratando algunas de sus características y funcionamiento en general. También se trataron las características de las herramientas y lenguajes utilizados en la creación de este tipo de aplicaciones. Lo anterior expuesto constituye los aspectos teóricos necesarios para luego dar solución al problema de la presente investigación.

Capítulo 2 Solución Propuesta

En el presente capítulo se proponen las soluciones técnicas en la creación de funcionalidades para la adquisición de información en Realidad Aumentada. Además se analiza la idea del componente a realizar y se dan los primeros pasos en su concepción práctica, desde el punto de vista de qué es lo que debe hacer el componente. Además se define la estructura principal del prototipo funcional elaborado y se explica de forma detallada la solución. También se especifican los requisitos funcionales y no funcionales, por último se lleva a cabo la expansión de los casos de uso.

2.1 Soluciones técnicas.

Se trabajará básicamente orientado a lograr un módulo multiplataforma que brinde las funcionalidades requeridas para la adquisición de videos desde distintas localizaciones, así como el trabajo sobre estos una vez en reproducción.

Después del análisis de las diferentes bibliotecas relacionadas con la RA, se ha planteado el uso de OpenCv ya que proporciona funciones para el procesado de imágenes a alto nivel y permite a los programadores crear aplicaciones poderosas en el dominio de la visión digital. Es un módulo multiplataforma, lo que posibilita que el módulo obtenido cumpla también con esta característica.

Entre las características deseables para implementar el módulo de adquisición de video, se partió de la base de que el lenguaje a utilizar debería ser orientado a objetos; pues es el paradigma más adecuado para la obtención de un módulo de calidad, reutilizable y extensible. Estos factores son muy importantes puesto que es probable que el componente pueda evolucionar y surjan necesidades o aportaciones futuras al mismo. Debe ser un lenguaje que permita utilizar la

Capítulo 2 Solución Propuesta

biblioteca OpenCV. Esto reduce drásticamente el abanico de posibilidades colocando a C++ como el lenguaje más adecuado para el desarrollo de este componente.

Las características más importantes de C++ son la programación orientada a objetos y la velocidad.

Para el proceso de desarrollo de la aplicación se seleccionó la metodología RUP, el lenguaje de modelado UML y la herramienta Visual Paradigm for UML 6.4 Enterprise Edition. Para más información ver epígrafes 1.6 y 1.7.3 del capítulo 1. Por último se optó por el compilador de Visual C++ que ofrece Microsoft Visual Studio.NET 2008 para la implementación. Para más información ver epígrafe 1.7.2 del capítulo 1.

Todo lo planteado posibilitó tener las herramientas necesarias para comenzar a desarrollar un prototipo funcional demostrativo de un módulo de video llamado *Módulo de adquisición de imágenes y video para herramienta de autoría de contenidos de realidad aumentada*.

2.2 Reglas del negocio.

1. Se podrán utilizar una o más cámaras.
2. Las cámaras se conectarán a la PC mediante USB.
3. Los videos puede tener cualquier dimensión (ancho por altura).
4. Se podrán reproducir tantos videos como se necesiten simultáneamente, sin importar la localización (archivo, red, cámara).
5. Los videos reproducidos desde archivo o red deben estar en formato AVI.
6. Leer ficheros de video implica solamente leer video, no incluye audio.

Capítulo 2 Solución Propuesta

2.3 Modelo de Dominio.

El diagrama de dominio tiene como objetivo principal identificar los conceptos más importantes y relacionarlos con el propósito de comprenderlos, estos se representan en el siguiente diagrama:

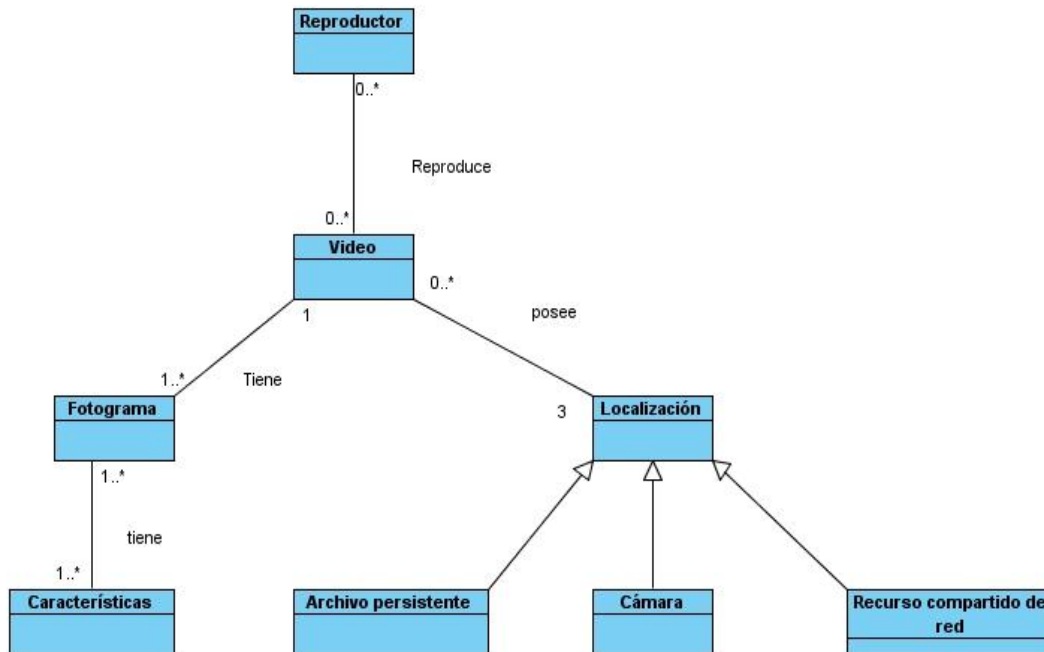


Figura 6: Modelo de Dominio.

En este modelo de dominio se representan los conceptos fundamentales asociados a la adquisición de video. Se representa el reproductor (que reproduce el video) y este tiene una localización determinada (archivo persistente, recurso compartido, cámara). Además, el video está compuesto por fotogramas, los cuales a su vez tienen distintas características.

2.3.1 Glosarios de Términos del Dominio.

Con el objetivo de lograr un mayor entendimiento del modelo de dominio, a continuación se enuncia el significado de los conceptos identificados en este:

Capítulo 2 Solución Propuesta

Reproductor: Programa o aplicación capaz de reproducir un video.

Video: Fichero con una extensión determinada que contiene datos de audio e imagen.

Fotograma: Tipo de información contenido en un fichero de video, se refiere al video.

Características: Información referente a alto, ancho, cantidad, velocidad del fotograma.

Localización: Referente a donde se encuentra el video.

Archivo persistente: Referente a localización, ubicado en el disco duro de una computadora.

Recurso compartido de red: Referente a localización, ubicado en un recurso compartido en una computadora conectada por la red.

Cámara: Referente a localización, dispositivo de captura de video, genera un video con la escena capturada.

2.4 Estructura principal de módulo.

En la figura 15 se aprecia la estructura principal del módulo de adquisición de video. La misma básicamente comprende 9 pasos distribuidos en 4 estados, inicialización, ciclo principal, operaciones y salida.

Capítulo 2 Solución Propuesta

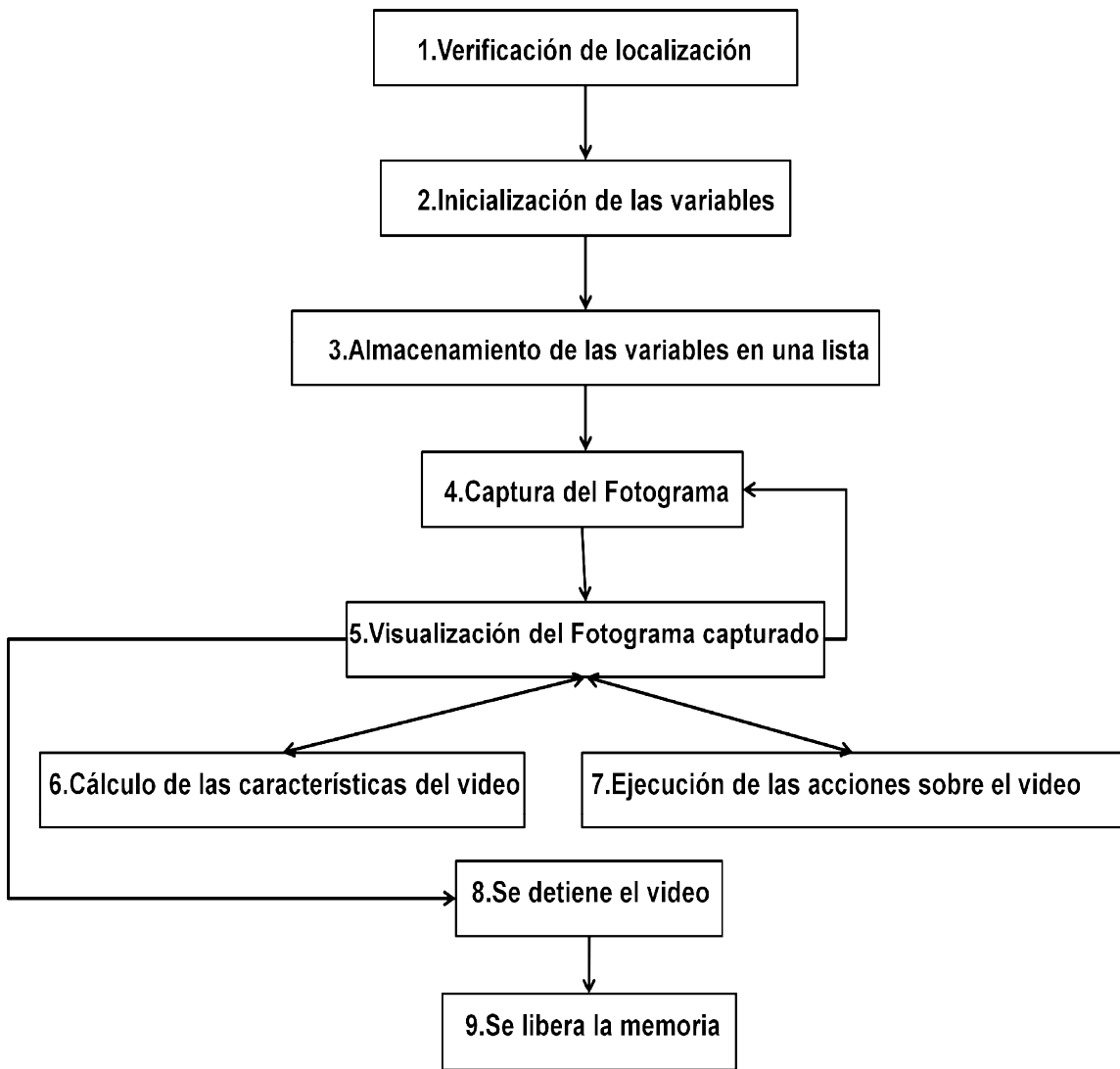


Figura 7: Estructura principal del módulo de adquisición de video.

- Inicialización :
El estado de inicialización contiene los tres primeros pasos. Primeramente se verifica si se reproducirá un video transmitido desde una cámara, ubicado en un archivo persistente en el disco duro o en un recurso compartido de red. En dependencia de lo anterior se inicializan

Capítulo 2 Solución Propuesta

las variables pertinentes y finalmente se almacenan en una lista para su posterior uso en los demás estados.

- **Ciclo Principal :**
Una vez ejecutados los tres primeros pasos, se pasa al segundo estado. En este estado se ejecuta el ciclo principal del componente y comprende los pasos 4 y 5. Aquí se captura cada fotograma del video y seguidamente se visualiza cada fotograma capturado.
- **Operaciones :**
En el tercer estado se van a llevar a cabo los pasos 6 y 7, los que consisten en calcular las características individuales de cada video en reproducción, así como de realizar distintas funcionalidades sobre los videos en reproducción (pausar, detener, moverse a un fotograma específico). En este estado solo se va a entrar si el usuario lo desea, en caso contrario se pasaría directamente al cuarto estado y final.
- **Salida :**
El cuarto estado está compuesto por los pasos 8 y 9. En estos pasos se procede a detener la captura y reproducción de video, así como cualquier otra acción que esté asociada a la terminación del programa. Posteriormente se liberan la memoria y recursos ocupados por las variables anteriormente inicializadas.

Los pasos 4 y 5 van a ser repetidos continuamente siempre que el programa esté en ejecución, mientras que los pasos 1, 2, 3 y 8, 9 se van a ejecutar al inicio y al final de la ejecución del programa, respectivamente. Además de los pasos especificados anteriormente, también se cuenta con los pasos 6 y 7 que solo se ejecutarán a solicitud del que interactúe con el componente.

2.5 Localización, captura, y reproducción del video.

Cuando trabajamos con video, debemos considerar diversas funciones, entre las que se encuentra como es lógico, la captura de ficheros de video. Este componente de video ofrece la posibilidad de cargar un video desde diferentes

Capítulo 2 Solución Propuesta

localizaciones, como son: archivo compartido de red, cámara, y disco duro. Para la captura del video en cada una de las localizaciones anteriormente mencionadas, el componente cuenta con los métodos *void InitCapture(char* window, int mode, char* dir , int Index)* y *void VideoRep()*, que se encargan de inicializar la captura teniendo en cuenta el modo de captura y posteriormente reproducir el video. Estos métodos serán explicados más detalladamente a continuación.

- Método *void InitCapture(char* window, int mode, char* dir , int Index)*:
Este método es uno de los más importantes en el componente ya que mediante él es que se va a inicializar las variables referentes al video. Las variables inicializadas en este método serán utilizadas posteriormente por el método *void VideoRep()* para la reproducción del video.
Como podemos observar a este método es necesario pasarle una serie de parámetros mediante los cuales se definen el modo, la ventana, y en caso de que sea necesario la dirección del video, o el identificador para la cámara. A continuación el autor explica estos parámetros más a fondo para su total entendimiento:
 - ✓ *char* window* : Mediante este parametro vamos a definir el nombre que tendrá la ventana en la que se abre el video.
 - ✓ *int mode* : Este parámetro es con el cual definimos si queremos abrir un video desde una cámara, o desde un archivo persistente o un recurso compartido. El parámetro será 1 si deseamos abrir un video ubicado en el disco duro o en un recurso compartido de red, o 2 si queremos reproducir un video desde una cámara en tiempo real.
 - ✓ *char* dir* : El parametro en que nos encontramos es referente a la dirección en la que se va a ubicar el video que necesitamos abrir y solo se especifica si anteriormente seleccionamos el modo 1 (abrir un video ubicado en el disco duro o en un recurso compartido de red). En caso de

Capítulo 2 Solución Propuesta

que hayamos seleccionado el modo 2 (reproducir un video desde una cámara en tiempo real) simplemente ponemos NULL.

Ejemplos:

"C:/OpenCV2.0/Build/samples/c/1x10.avi"

"//10.35.17.252/d\$/Alejandro/1x10.avi"

- ✓ *int Index* : El uso de este parametro se limita al caso en que hayamos seleccionado el modo 2 (reproducir un video desde una cámara en tiempo real). Este parámetro es un identificador que será 0 si solo tenemos una cámara e incrementa cuando hay múltiples cámaras en el mismo sistema. La otra parte del identificador es el dominio de la cámara e indica en esencia qué tipo de cámara tenemos. El dominio puede ser cualquiera de las constantes predefinidas, mostradas en la siguiente tabla:

Constante de Captura para la Cámara	Valor numérico
CV_CAP_ANY	0
CV_CAP_MIL	100
CV_CAP_VFW	200
CV_CAP_V4L	200
CV_CAP_V4L2	300
CV_CAP_FIREWIRE	300
CV_CAP_IEEE1394	300
CV_CAP_DC1394	300
CV_CAP_CMU1394	300

Tabla 1: Distintos tipos de dominios para cámaras.

El identificador final viene siendo la suma del dominio y el identificador de la cámara. En la mayoría de los casos el dominio es innecesario cuando tenemos solo una cámara. En caso de haber seleccionado el modo 1

Capítulo 2 Solución Propuesta

(abrir un video ubicado en el disco duro o en un recurso compartido de red) no es necesario especificar nada en ese campo.

- Método *void VideoRep()* :

Este método también consta de una gran importancia en el componente ya que es el encargado de la reproducción del video. No necesita parámetros ya que trabaja con los parámetros especificados en el método anterior. Básicamente lo que hace es un ciclo en el cual se va mostrando fotograma a fotograma del video, hasta que este termine o el usuario decida detener la reproducción.

2.6 Operaciones sobre el video.

Una vez cargado el video, el componente brinda la posibilidad de interactuar con el video en reproducción. Tenemos a nuestra disposición diversas operaciones que se pueden realizar sobre el video ya sea para obtener sus características individuales o para realizar distintas funciones sobre el mismo. Para esto el componente cuenta con métodos como *int FotogramaActualEnReproduccion()* y *void Move()* que pueden ser utilizados por todo el que incluya este componente para lograr las funcionalidades que desee. En los siguientes sub epígrafes serán explicados estos métodos más detalladamente.

2.6.1 Obtención de las características del video.

Para obtener todo lo referente a características y propiedades del video, así como el comportamiento en tiempo de reproducción, podemos apoyarnos en distintos métodos con los que cuenta el componente. A continuación explicamos la utilidad de algunos de estos métodos.

- *int FotogramaActualEnReproduccion()*: Con la llamada a este método vamos a obtener un dato de tipo entero que nos va a indicar el

Capítulo 2 Solución Propuesta

fotograma que se está reproduciendo en ese momento (en tiempo de ejecución).

- *int CantidadTotalFotogramas()* : Con la llamada a este método vamos a obtener un dato de tipo entero que básicamente nos va a indicar la cantidad total de fotogramas con los que cuenta el video en reproducción. Claro está, que este método solo se podrá ejecutar satisfactoriamente cuando se trate de un video existente en un archivo persistente en el disco duro, o en un recurso compartido en la red. Ya que si es una reproducción desde una cámara este dato sería imposible de calcular.
- *int VelocidadFxS()*: Al llamar a este método obtendremos un tipo de dato entero, referente a velocidad a la que fue capturado el video en reproducción (expresada en fotogramas por segundo). Este dato puede ser importante si queremos reproducir este video a una velocidad óptima.
- *int AnchoVideo()*: Con la llamada a este método se obtiene un dato de tipo entero que nos indicará el ancho de los fotogramas que componen el video en reproducción.
- *int AltoVideo()*: Con la llamada a este método se obtiene un dato de tipo entero que nos indicará el alto de los fotogramas que componen el video en reproducción.

2.6.2 Funcionalidades a realizar sobre el video.

Cuando el video sobre el que estamos trabajando está ubicado en un recurso compartido en la red o en un archivo persistente en el disco duro, podemos realizar diversas funcionalidades sobre este, llamando distintos métodos con los que cuenta el componente. A continuación explicamos algunos de estos métodos.

- *Void Play()*: Este método básicamente lo que hace es poner en falso las variables booleanas stop y pause, y en verdadero la variable *booleana*

Capítulo 2 Solución Propuesta

play dando paso a la reproducción del video explicada en el método *void VideoRep()*.

- *Void Pause()*: Este método pone en falso las variables booleanas *stop* y *play*, y en verdadero la variable booleana *pause*, haciendo una pausa en el método *void VideoRep()* hasta que vuelva a ser llamado el método *void Play()*.
- *Void Stop()*: Este método básicamente lo que hace es poner en falso las variables booleanas *play* y *pause*, y en verdadero la variable *stop*. Posibilita que de esta forma en el método *void VideoRep()* la reproducción sea detenida y movida al fotograma 0.
- *Void Forward()* y *void Backward()*: Como sus nombres lo indican, estos métodos dan la posibilidad de moverse hacia adelante o hacia atrás en el video, básicamente lo que hacen es aumentar o disminuir el fotograma en que se encuentra la reproducción actual, haciendo que se adelante o se atrase la reproducción.
- *Void Moverse()*: Mediante este método es posible moverse a cualquier fotograma del video, simplemente le pasamos por consola el fotograma al que deseemos movernos y automáticamente la reproducción se mueve hacia ese fotograma, esto se consigue modificando el fotograma en que se encuentre la reproducción. Para lograr esto también existe la barra de desplazamiento pero el método *void Moverse()* es más exacto.

2.7 Utilidad.

La elaboración de este componente de video será de gran utilidad para la herramienta de autoría de contenidos de Realidad Aumentada ya que con su inclusión a este proyecto se logrará la adquisición de video desde diversas fuentes (archivo, dispositivo de captura, recurso compartido de red), así como reproducción de varias cámaras simultáneamente entre otras funcionalidades. Este componente también podrá ser utilizado en cualquier aplicación de Realidad Virtual o Aumentada que necesite adquirir video, simplemente tiene

Capítulo 2 Solución Propuesta

que agregar el componente como una biblioteca más y así tendrá acceso a todos sus métodos.

2.8 Captura de requisitos.

La captura de requisitos es una de las actividades del proceso de desarrollo de requisitos. En esta actividad se identifican las capacidades, condiciones y restricciones que debe tener un software. A continuación se hace referencia a los requisitos funcionales y no funcionales más significativos capturados para el desarrollo de la biblioteca.

2.8.1 Requisitos funcionales.

RF1. Cargar video desde una localización.

RF1.1 Cargar video desde una cámara web.

RF1.2 Cargar video desde archivo persistente en el disco duro.

RF1.3 Cargar video desde recurso compartido de red.

RF2. Reproducir uno o varios videos al mismo tiempo sin importar localización de estos.

RF3. Calcular características de los videos en reproducción.

RF3.1 Calcular alto de los fotogramas de los videos.

RF3.2 Calcular ancho de los fotogramas de los videos.

RF3.3 Calcular cantidad total de fotogramas que posee el(los) video(s).

RF3.4 Calcular velocidad en fotogramas por segundo a la que fue capturado el(los) video(s).

RF3.5 Calcular el fotograma actual en reproducción en el(los) video(s).

Capítulo 2 Solución Propuesta

RF4. Realizar funcionalidades sobre el(los) video(s) en reproducción.

RF4.1 Reproducir video(s).

RF4.2 Pausar reproducción.

RF4.3 Detener reproducción.

RF4.4 Mover reproducción a un fotograma especificado.

RF4.5 Mover reproducción hacia adelante.

RF4.6 Mover reproducción hacia atrás.

2.8.2 Requisitos no funcionales.

Usabilidad

- El modulo elaborado podrá ser utilizado por cualquier aplicación que necesite adquirir video.

Portabilidad

- Multiplataforma (integrarse a entornos de desarrollo en Linux o Windows).

Rendimiento

- Debe ser capaz de reproducir varios videos al mismo tiempo sin alterar la velocidad de reproducción original de cada uno de estos videos.

Hardware

- Es necesaria una PC con CPU Intel Pentium IV, 1 Gb de HDD, 256 Mb de memoria RAM.

Software

- Es necesario utilizar OpenCV.

Capítulo 2 Solución Propuesta

2.9 Diagrama de casos de uso del sistema.

Se propone el diagrama de casos de uso que representa la figura 16, el cual agrupa todos los requisitos funcionales.

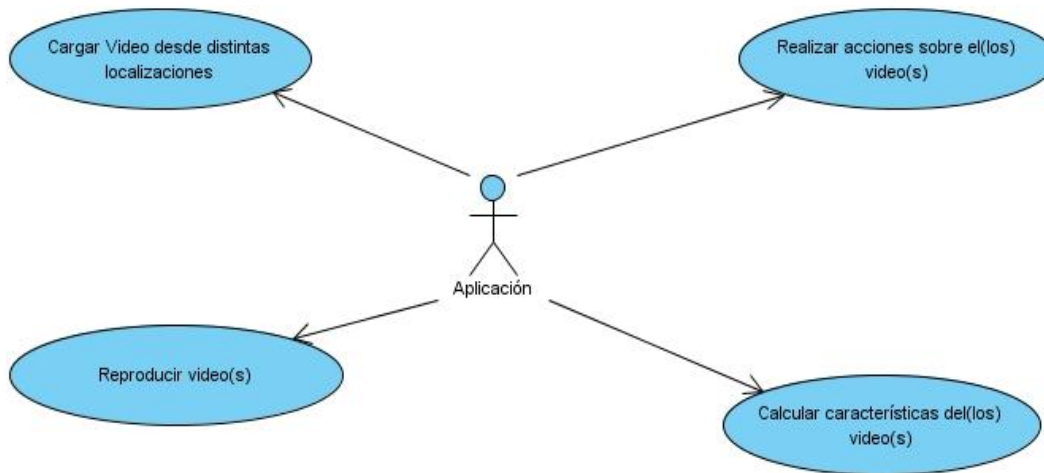


Figura 8: Diagrama de casos de uso.

2.9.1 Definición del actor del sistema.

Actores	Justificación
Aplicación	La aplicación que utilice la biblioteca es la que se beneficiará con las funcionalidades que brinda este módulo para adquisición de video.

Tabla 2: Definición de los actores.

2.9.2 Expansión de los casos de uso.

Nombre del caso de uso	Cargar video desde distintas localizaciones
Actores	Aplicación
Propósito	Cargar el fichero de video en dependencia de la localización y almacenar la información referente a este en una lista para su

Capítulo 2 Solución Propuesta

	posterior utilización.
<p>Resumen: El caso de uso se inicia cuando la Aplicación solicita cargar un video y se llaman los métodos predefinidos para cargar la información del video en cada uno de los casos definidos para las distintas localizaciones, entonces se procede al almacenamiento de estos en una lista para su posterior utilización en la reproducción del mismo. El caso de uso termina cuando la información del video es cargada y almacenada.</p>	
Precondiciones	Que exista un fichero de video o una cámara mediante la cual se pueda reproducir el video.
Referencias	RF1, RF1.1, RF1.2, RF1.3
Curso normal de los eventos	
Acción del actor	Respuesta del sistema
1- Solicita cargar un video desde un archivo persistente en el disco duro o recurso compartido de red, especificandola dirección	<p>1.1- Se llama a los métodos predefinidos en el módulo que se emplearán para cargar la información del video.</p> <p>1.2- Se cargan los datos del video.</p>
2- Solicita cargar un video desde una cámara especificando el identificador.	<p>2.1- Se llama a los métodos predefinidos en el módulo que se emplearán para cargar la información del video.</p> <p>2.2- Se cargan los datos del video.</p>
Curso Alternativo de los eventos	
	<p>1.1- Si el fichero de video no existe en la dirección entrada se muestra un mensaje de error y termina el caso de uso.</p> <p>2.1- Si el sistema no reconoce ninguna cámara conectada, se muestra un mensaje de error y termina el caso de uso.</p>
Post condiciones	La información referente al video es almacenada en una lista para su posterior

Capítulo 2 Solución Propuesta

	utilización.
--	--------------

Tabla 3: Expansión del Caso de Uso (Cargar video desde distintas localizaciones).

Nombre del caso de uso	Reproducir video(s)	
Actores	Aplicación	
Propósito	Reproducir uno o varios videos sin importar la localización.	
Resumen: El caso de uso se inicia cuando la Aplicación solicita reproducir un video, y se llaman los métodos predefinidos para reproducir la información del video almacenada previamente en una lista. El caso de uso termina a petición de la aplicación o cuando el video termina de reproducirse.		
Precondiciones	Que la información referente al video este previamente almacenada.	
Referencias	RF2	
Curso normal de los eventos		
Acción del actor	Respuesta del sistema	
1- Solicita reproducir un video.	1.1- Se inicializan los sistemas que se emplearán para reproducir la información del video previamente almacenada. 1.2- Se inicia la reproducción del video.	
Curso Alternativo de los eventos		
Post condiciones		

Tabla 4: Expansión del Caso de Uso (Reproducir video(s)).

Nombre del caso de uso	Calcular características del(los) video(s)	
Actores	Aplicación	
Propósito	Calcular distintas características al(los) video(s) en reproducción.	

Capítulo 2 Solución Propuesta

Resumen: El caso de uso se inicia cuando la Aplicación solicita calcular las características del(los) video(s) en reproducción, y se llaman los métodos definidos para calcular las distintas características. El caso de uso termina cuando son calculadas dichas características.	
Precondiciones	Que haya al menos un video en reproducción.
Referencias	RF3, RF3.1, RF3.2, RF3.3, RF3.4, RF3.5
Curso normal de los eventos	
Acción del actor	Respuesta del sistema
1- Solicita calcular las características de los videos en reproducción.	1.1- Se llaman los métodos predefinidos, encargados de calcular las características del(los) video(s) en reproducción. 1.2- Se calculan dichas características.
Curso Alternativo de los eventos	
Post condiciones	

Tabla 5: Expansión del Caso de Uso (Calcular características del(los) video(s)).

Nombre del caso de uso	Realizar funcionalidades sobre el(los) video(s)
Actores	Aplicación
Propósito	Realizar distintas funcionalidades sobre el(los) video(s) en reproducción.
Resumen: El caso de uso se inicia cuando la Aplicación solicita realizar distintas funcionalidades sobre el(los) video(s) en reproducción, se llaman los métodos predefinidos para realizar las distintas funcionalidades. El caso de uso termina cuando la aplicación termina de realizar las funciones.	
Precondiciones	Que haya al menos un video en reproducción.
Referencias	RF4, RF4.1, RF4.2, RF4.3, RF4.4, RF4.5, RF4.6
Curso normal de los eventos	
Acción del actor	Respuesta del sistema

Capítulo 2 Solución Propuesta

1- Solicita calcular las características del(los) video(s) en reproducción.	1.1- Se inicializan los sistemas que se emplearán para realizar las funcionalidades sobre el(los) video(s) en reproducción. 1.2- Se realizan dichas funcionalidades.
Curso Alternativo de los eventos	
Post condiciones	

Tabla 6: Expansión del Caso de Uso (Realizar funcionalidades sobre el(los) video(s)).

En este capítulo se detalló la solución propuesta exponiendo la estructura que tiene la misma. Luego se hizo hincapié en la captura y localización de video, especificando las distintas ubicaciones desde las que se podría cargar y la manera en la que habría que proceder en cada uno de estos casos. Estos aspectos constituyeron la base para así lograr la obtención de las características del video y las funcionalidades que se pueden realizar sobre el mismo mediante distintos métodos. Se establecieron las capacidades o funcionalidades que el sistema debe cumplir y las propiedades o cualidades que el producto debe tener, es decir, los requisitos funcionales y no funcionales. Se agruparon los requisitos funcionales en casos de uso y estos últimos se describieron para un mejor entendimiento de los procesos que tendrán lugar en el producto desarrollado.

Capítulo 3 Diseño e implementación

Capítulo 3: Diseño e implementación.

En este capítulo se tratan los aspectos fundamentales del diseño e implementación de la solución para la elaboración del módulo de adquisición de video. Se muestra el diseño de clases del sistema propuesto, en donde se definen las responsabilidades de estas y sus relaciones. Además se muestran los diagramas de secuencia por casos de uso que intervendrán en el ciclo de desarrollo del módulo, facilitando el entendimiento de este.

3.1 Diagrama de clases de diseño.

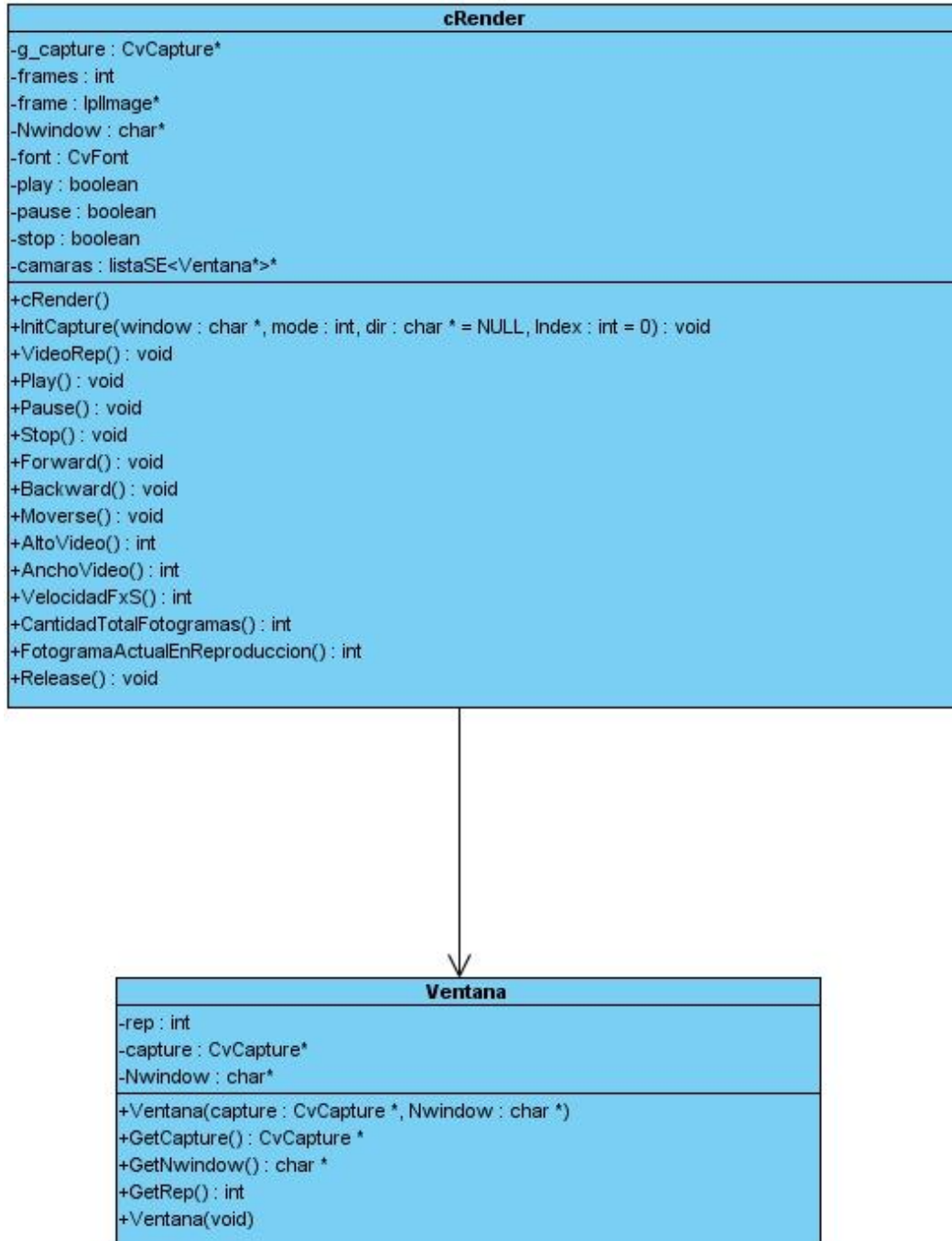


Figura 9: Diagrama de clases de diseño.

Capítulo 3 Diseño e implementación

3.2 Descripción de las clases.

Nombre de la clase: cRender	
Tipo de clase: Controladora	
Atributo	Tipo
g_capture	CvCapture*
frames	int
frame	IplImage*
Nwindow	char*
font	CvFont
play	boolean
pause	boolean
stop	boolean
camaras	listaSE<Ventana*>*
Responsabilidades de la clase:	
Nombre:	cRender()
Descripción:	Constructor de la clase.
Nombre:	InitCapture(window : char *, mode : int, dir : char* = NULL, Index : int = 0) : void
Descripción:	Se selecciona el modo (referente a la localización del video) y posteriormente se inicializan las variables referentes al video (ventana y dirección o index en sus respectivos casos). Luego son almacenados para su posterior uso en el método VideoRep().
Nombre:	VideoRep() : void
Descripción:	Consta del ciclo principal de la aplicación en el cual se va mostrando fotograma a fotograma del video, hasta que éste termine o el usuario decida detener la reproducción. Trabaja con las variables inicializadas en el método void InitCapture(window : char *, mode : int, dir : char* = NULL, Index : int = 0).

Capítulo 3 Diseño e implementación

Nombre:	Play() : void
Descripción:	Se pone la variable bool play en verdadero y se llama a los métodos encargados de la reproducción del video.
Nombre:	Pause() : void
Descripción:	Se pone la variable bool pause en verdadero y se manda a los métodos encargados de la reproducción a pausar el video.
Nombre:	Stop() : void
Descripción:	Se pone la variable bool stop en verdadero y se llama a los métodos encargados de la reproducción del video.
Nombre:	Forward() : void
Descripción:	Se adelanta la reproducción del video, modificando el fotograma actual en reproducción.
Nombre:	Backward() : void
Descripción:	Se atraza la reproducción del video modificando el fotograma actual en reproducción.
Nombre:	Moverse() : void
Descripción:	Se mueve la reproducción del video a un fotograma determinado modificando el fotograma actual en reproducción.
Nombre:	AltoVideo() : int
Descripción:	Se obtiene un entero referente al alto de los fotogramas del video en reproducción.
Nombre:	AnchoVideo() : int
Descripción:	Se obtiene un entero referente al ancho de los fotogramas del video en reproducción.
Nombre:	VelocidadFxS() : int
Descripción:	Se obtiene un entero referente a la velocidad en fotogramas por segundo a la que fue capturado el video en reproducción.
Nombre:	CantidadTotalFotogramas() : int
Descripción:	Se obtiene un entero referente a la cantidad total de fotogramas que contiene el video en reproducción.

Capítulo 3 Diseño e implementación

Nombre:	FotogramaActualEnReproduccion() : int
Descripción:	Se obtiene un entero referente al fotograma actualmente en reproducción.
Nombre:	Release() : void
Descripción:	Son destruidas las variables y es liberado el espacio ocupado en memoria.

Tabla 7: Descripción de la clase (cRender).

Nombre de la clase: Ventana	
Tipo de clase: Entidad	
Atributo	Tipo
rep	int
capture	CvCapture*
Nwindow	char*
Responsabilidades de la clase:	
Nombre:	Ventana(capture : CvCapture *, Nwindow : char *)
Descripción:	Constructor de la clase.
Nombre:	GetCapture() : CvCapture*
Descripción:	Retorna el valor de la variable CvCapture* capture.
Nombre:	GetNwindow() : char*
Descripción:	Retorna el valor de la variable char* Nwindow.
Nombre:	GetRep() : int
Descripción:	Retorna el valor de la variable intrep.
Nombre:	~Ventana(void)
Descripción:	Destructor de la clase. Libera el espacio ocupado en memoria.

Tabla 8: Descripción de la clase (Ventana).

Capítulo 3 Diseño e implementación

3.3 Diagramas de Secuencia.

En este epígrafe se muestran los diagramas de secuencia correspondientes a los casos de uso descritos anteriormente en el capítulo 2. Estos diagramas representan los eventos generados por el actor, su orden y los eventos internos del sistema. Para lograr claridad en los diagramas se detallaron los eventos de mayor importancia.

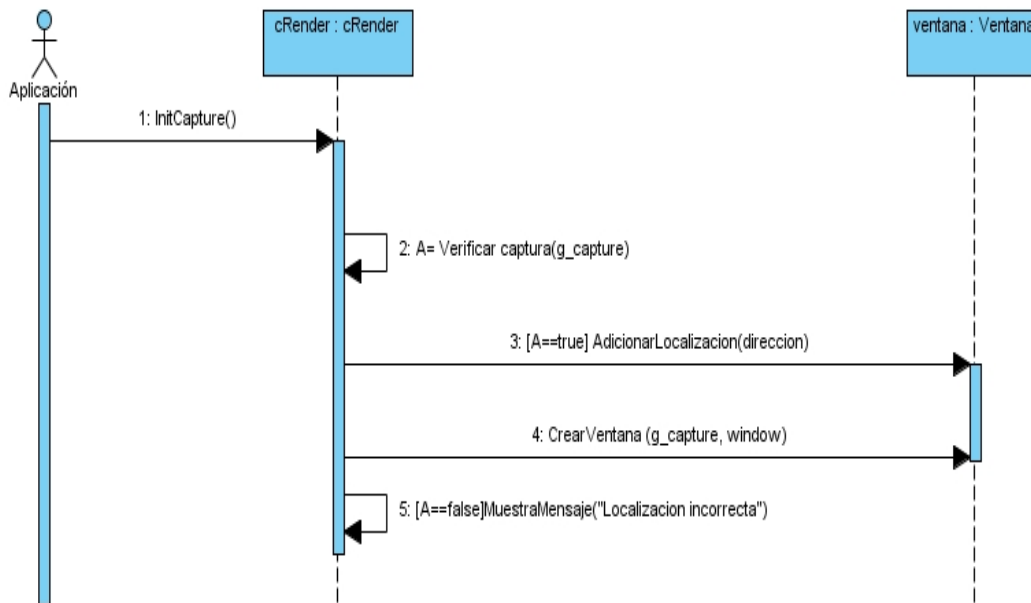


Figura 10: Diagrama de secuencia del diseño. Cargar video desde distintas localizaciones.

Capítulo 3 Diseño e implementación

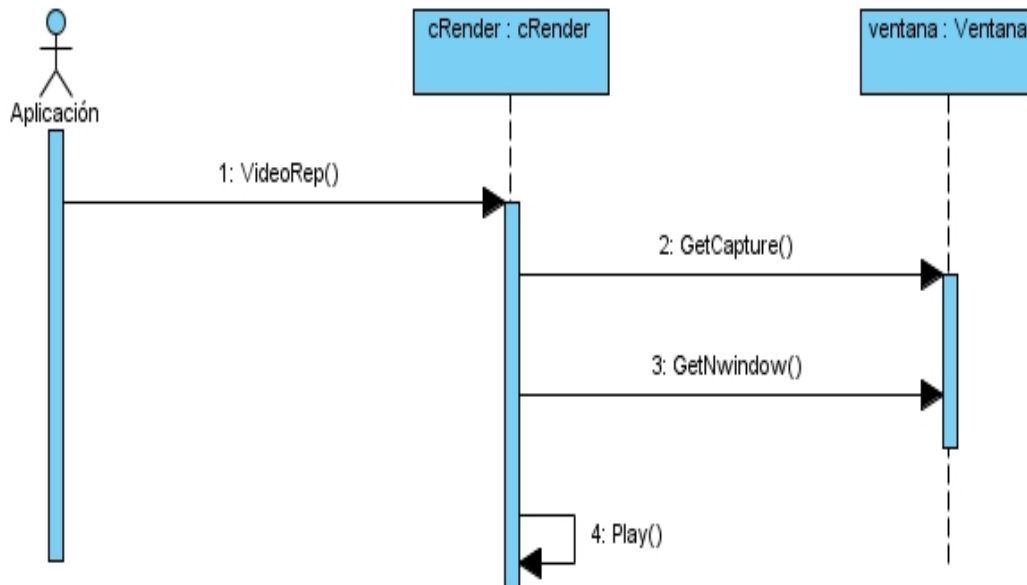


Figura 11: Diagrama de secuencia del diseño. Reproducir video.

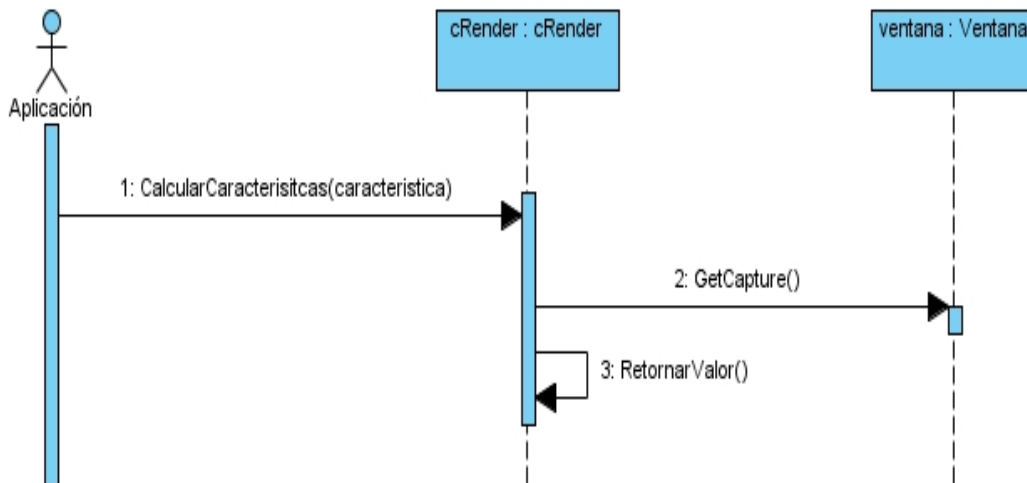


Figura 12: Diagrama de secuencia del diseño. Calcular características del(los) video(s).

En el diagrama de secuencia anterior se generalizan los métodos AltoVideo(), AnchoVideo(), VelocidadFxS(), CantidadTotalFotogramas(), FfotogramaActualEnReproduccion() en el método CalcularCaracteristicas(característica) ya que en todos se procede de la misma forma.

Capítulo 3 Diseño e implementación

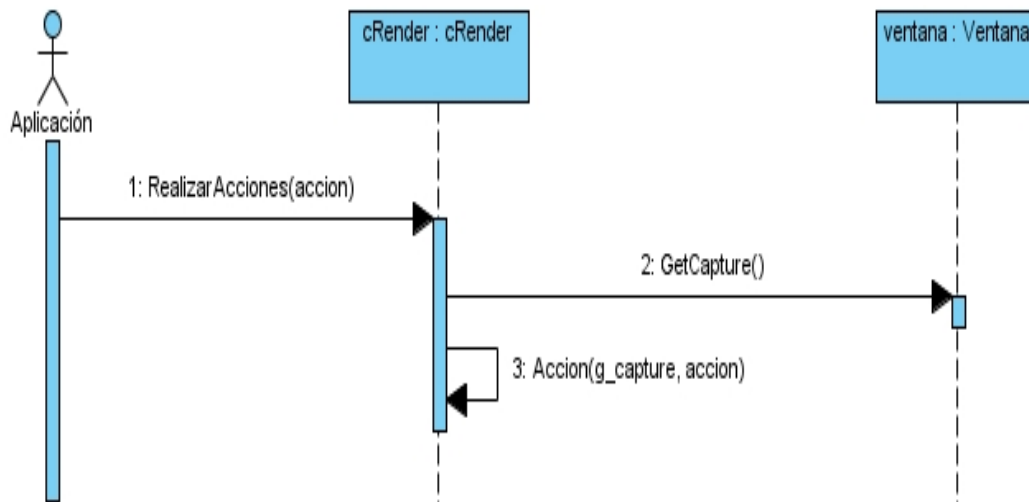


Figura 13: Diagrama de secuencia del diseño. Realizar acciones sobre el(los) video(s).

En el diagrama de secuencia anterior se generalizan los métodos Pause(), Stop(), Play(), Moverse(), Forward(), Backward() en el método RealizarAcciones(acción) ya que en todos se procede de la misma forma.

3.4 Diagrama de componentes.

La transformación de las clases de la lógica a su forma física se hizo mediante la utilización de componentes. Los componentes utilizados para la elaboración de la aplicación se agruparon en paquetes. El paquete VideoWork es el que agrupa los componentes definidos para la elaboración del prototipo funcional y en el paquete OpenCV están los componentes específicos de dicha biblioteca.

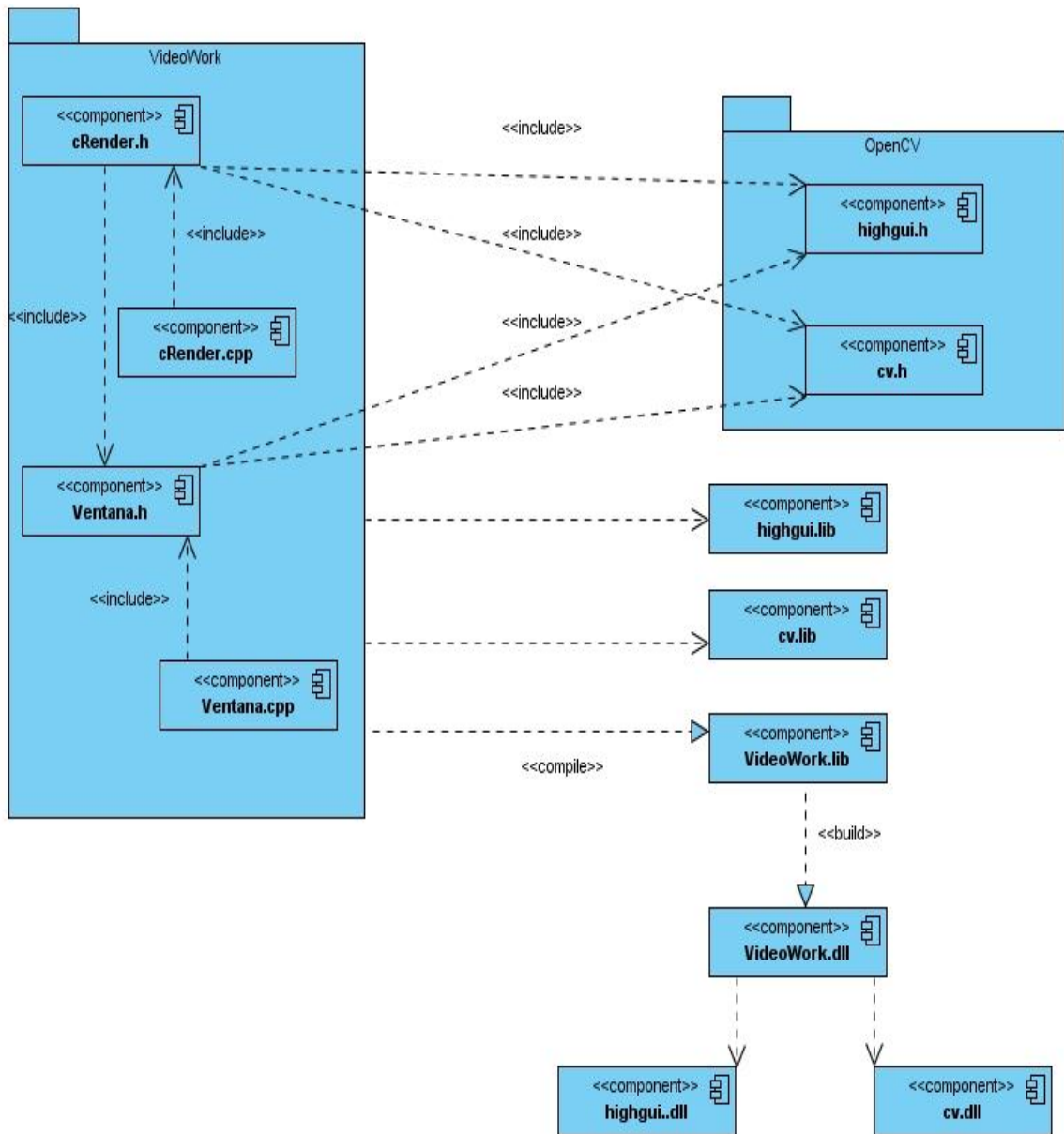


Figura 14: Diagrama de componentes.

3.5 Diagrama de despliegue.

En la siguiente figura se muestra el diagrama de despliegue. Aunque este es un diagrama sencillo se decidió incluirlo para enfatizar en la posibilidad del uso de una o más cámaras conectadas a la computadora, cumpliendo de esta forma

Capítulo 3 Diseño e implementación

con uno de los requisitos más importantes de la investigación (adquisición de video desde varias cámaras simultáneamente).

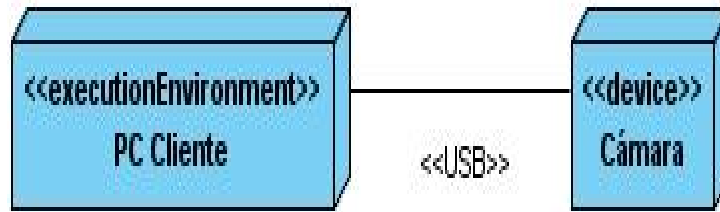


Figura 15: Diagrama de despliegue.

3.6 Discusión de los resultados.

Con esta investigación se logró un componente, capaz de obtener videos simultáneamente desde distintas localizaciones, cumpliendo de esta forma con su objetivo general. La velocidad de reproducción de los videos no se ve afectada por la cantidad de videos que se encuentren en reproducción, esta será la misma en cualquier caso. En la tabla 9 se muestra la velocidad de reproducción del módulo para uno y más videos desde distintas localizaciones, expresada en fotogramas por segundo.

Cant de videos en rep.	Localización	Velocidad (F/s)
1	Cámara	30
1	Archivo persistente	30
1	Recurso compartido de red	30
3	Cámara, archivo persistente y recurso compartido de red.	30
6	Cámara, archivo persistente y recurso compartido de red.	30

Tabla 9: Velocidad de reproducción del módulo para uno y más videos.

Capítulo 3 Diseño e implementación

En este capítulo se realizó el diseño e implementación del prototipo funcional. Se obtuvo el diagrama de clases del diseño, las cuales fueron descritas posteriormente con sus atributos y responsabilidades principales para lograr un mayor entendimiento. También se obtuvieron los diagramas de secuencia correspondientes a los casos de uso descritos en el capítulo 2. Posteriormente se obtuvieron el diagrama de componente y de despliegue correspondiente al flujo de trabajo de implementación, lo cual facilitó la codificación de la solución. Por último se hizo una discusión de los resultados para resaltar algunos aspectos de rendimiento.

Conclusiones

Conclusiones

Con el trabajo se obtuvo una biblioteca capaz de adquirir video desde diversas localizaciones (archivo persistente, recurso compartido de red y dispositivo de captura), utilizando la biblioteca OpenCv para leer y reproducir dichos videos; además de permitir visualizar más de un video a la vez, así como reproducir varias cámaras simultáneamente.

Se logró una biblioteca que provee una serie de funcionalidades que abstraen al programador al nivel de solo tener que especificar el video que se desea mostrar y la localización del mismo. Además de poder calcular algunas de sus características y realizar diferentes acciones sobre el mismo.

Recomendaciones

Recomendaciones

Una vez concluida la investigación con un resultado satisfactorio, se recomienda que esta biblioteca obtenida se utilice para la adquisición de video en Sistemas de Realidad Aumentada, dadas su facilidad de utilización. También se recomienda:

- Seguir desarrollando este componente para que en un futuro soporte más formatos de video.
- Incorporar la funcionalidad de reproducir video desde cámaras IP o AXIS.

Referencias Bibliográficas

Referencias Bibliográficas

1. Luis. monografias.com. *monografias.com*. [Online] [Cited: 12 10, 2010.]
<http://www.monografias.com/trabajos10/vire/vire.shtml..>
2. **Alfaro Ferreres, Luis and Roca Estellés, María José.** Universidad de Jaén. *Universidad de Jaén*. [Online] 12 2010.
<http://wwdi.ujaen.es/asignaturas/informatica/Teoria/Software%20Grafico/Graficodoc/Graficos%20audio%20y%20video.doc..>
3. Megaservice A.S.D.A S.R.L. *Megaservice A.S.D.A S.R.L.* [Online] 12 2010.
http://www.megaservice.com.ar/Info_espe/Los%20archivos%20AVI.htm..
4. **AZUMA.** 1997.
5. 2005, Revista Universidad EAFIT.
6. informaticamoderna.com. *informaticamoderna.com*. [Online] 2011.
http://www.informaticamoderna.com/Camara_web.htm.
7. electricBricks. *electricBricks*. [Online] 2011.
<http://blog.electricbricks.com/es/2010/05/instalacion-opencv/>.
8. nelsoncortesdigital.blogspot.com. *nelsoncortesdigital.blogspot.com*. [Online] 2010.
[http://nelsoncortesdigital.blogspot.com/2010/11/opencv-libreria-para-la-
implementacion..](http://nelsoncortesdigital.blogspot.com/2010/11/opencv-libreria-para-la-implementacion..)
9. ARToolKit. *ARToolKit*. [Online] 2010. <http://www.hitl.washington.edu/artoolkit/>.
10. EZFLAR. *EZFLAR*. [Online] 2009. http://www.ezflar.com/home/show_home.
11. words.transmote.com. *words.transmote.com*. [Online] 2010.
<http://words.transmote.com/wp/flarmanager/>.

Referencias Bibliográficas

12. **Booch, Wesley, Addison.***El Lenguaje Unificado de Modelado*. 1998.
13. es.answers.yahoo.com. *es.answers.yahoo.com*. [Online] 2008.
<http://es.answers.yahoo.com/question/index?qid=20070811200816AAA6CvJ>.
14. **Jimenez, Alex.** alexjimenez. *alexjimenez*. [Online] 11 29, 2007.
<http://alexjimenez.wordpress.com/2007/11/29/caracteristicas-de-visual-studio-2008-y-sql-server-2008/>.
15. Free Download Manager. *Free Download Manager*. [Online] Mar 05, 2007.
[http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(M%C3%8D\)_14720_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p/).
16. **Azuma.** 2001.
17. **Pablo Acevedo y Maxi Alcanis.** [Online] Octubre 2010.
<http://apasionadosporlahistoria.blogspot.com/2010/10/realidad-aumentada-ra.html>.
18. **AZUMA.***A Survey of Augmented Reality*. 1997. Vol. 4, [Consultado en: 12-12-2007].
Disponible en: <http://www.cs.unc.edu/~azuma/ARpresence.pdf>.
19. **O. Bimber and R. Raskar.***Spatial Augmented Reality: Merging Real and Virtual Worlds*, A.K Peters. 2005.
20. **M. Billinghurst and H. Kato.***Collaborative Augmented Reality*. 2002. pp. 64-70. Vol. 7.
21. **Azuma ,Ronald,Yohan Baillot, Reinhold Behringer, Steven Feiner, Simon Julier, Blair MacIntyre.***Recent Advances in Augmented Reality. IEEE Computer Graphics and Applications*. 2001. Vol. 6 , [Consultado en: 16-11-2007]. Disponible en:
<http://www.cs.unc.edu/~azuma/cga2001.pdf>.
22. *AR-Immersive Cinema at the Aula Natura Visitors Center.* **Zheng, Wenjun.**
Octubre_Diciembre 2010.

Referencias Bibliográficas

23. mailxmail.com. *mailxmail.com*. [Online] 2011. <http://www.mailxmail.com/cursos-introduccion-gps>.
24. **Héctor López Pombo**. *Análisis y desarrollo de Sistemas de Realidad Aumentada*. 2009-2010. Informe de tesis de maestría.
25. profesormolina.com. *profesormolina.com*. [Online] 2011. http://www.profesormolina.com.ar/tecnologia/sens_transduct/tipos.htm.
26. BALLUFF. *BALLUFF*. [Online] 2011. <http://www.balluff.com/Balluff/Website/Templates/Products/ProductGroupItem.aspx?NRMODE=Published&NRNODEGUID={049E9CDE-E215-43F7-A6E8-DAE09EB6C125}&NRORIGINALURL=/Balluff/es/ProductsChannel/Overview/es/Sensores%20bme68%20C3%20A1nicos.htm&NRCACHEHINT=Guest>.
27. profesormolina.com.ar. *profesormolina.com.ar*. [Online] 2011. http://www.profesormolina.com.ar/tecnologia/sens_transduct/sensores.htm.
28. ucm.es. *ucm.es*. [Online] 2011. http://www.ucm.es/info/otri/complutecno/fichas/tec_evernabeu2.htm.
29. **Booch**. *El Lenguaje Unificado de Modelado*. Addison Wesley, 1998. p. 1998.

Anexos

Anexos

Anexo1

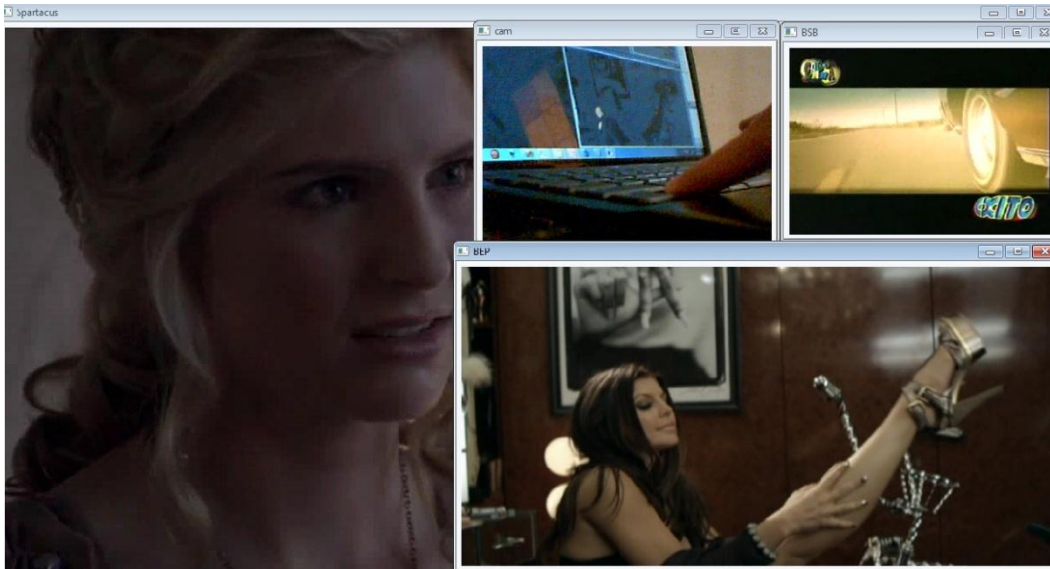


Figura 16: Adquisición de varios videos de archivo, red y cámara.

Anexo2

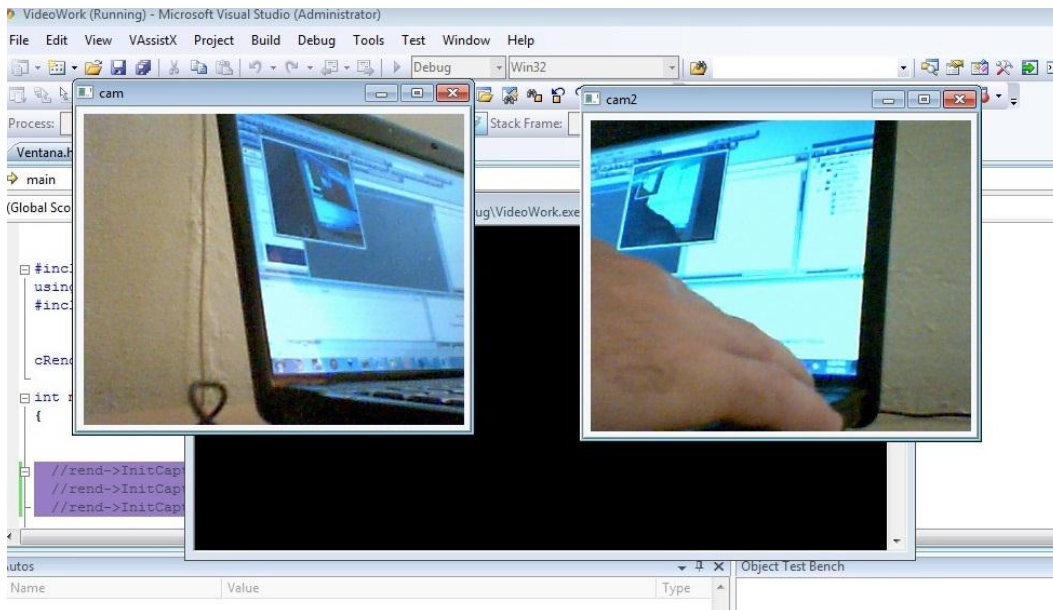


Figura 17: Adquisición de video desde dos cámaras simultáneamente.

Anexos

Anexo 3



Figura 18: Cálculo de las características de un video.

Anexo 4

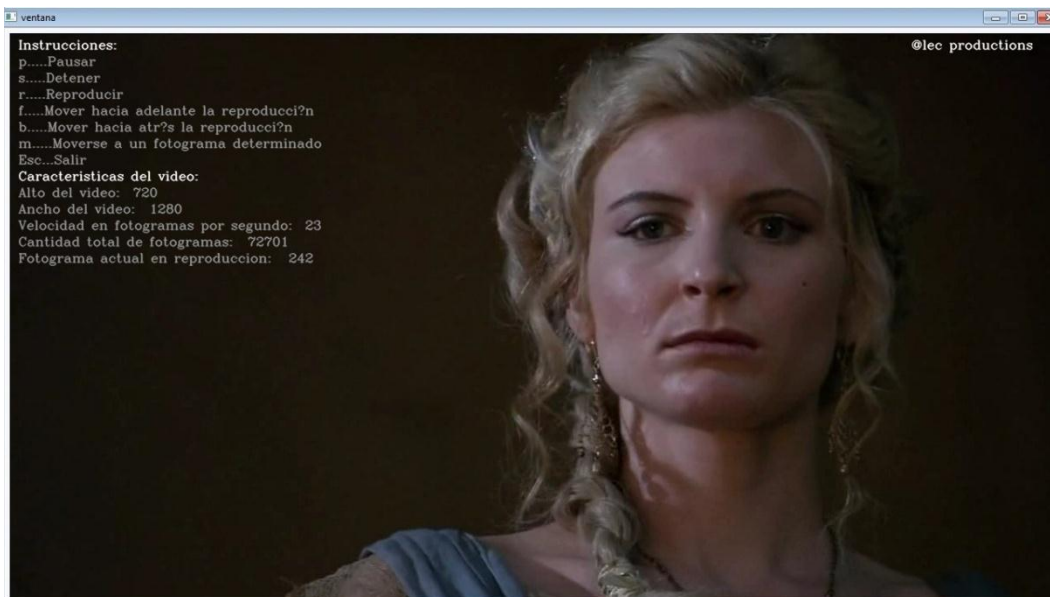


Figura 19: Realización de funciones sobre un video.

Anexos

Anexo5



Figura 20: Adquisición de video desde un archivo

Glosario de Términos

Glosario de Términos

Archivo: Elemento en el que se guarda información: textos, datos, gráficos, fotos, películas, música. Se identifica mediante un nombre, por ejemplo, "misdatos.txt", "foto.jpg", "cancion.mp3" o "video.avi".

AVI: Es un tipo de archivo contenedor, que puede contener una pista de video codificada y una o más pistas de audio. Su nombre proviene del hecho de que el video y audio están "entrelazados" dentro del contenedor.

C++: Lenguaje de programación orientado a objetos.

Codec: Acrónimo de codificador/decodificador. Es un software que consta de una serie de algoritmos que permiten codificar y decodificar contenidos multimedia.

Digital: Representación de la información mediante combinaciones de unidades binarias o 'bits'.

Formato: En el mundo de la informática, es el conjunto de reglas o especificaciones mediante las cuales se pueden organizar datos de diversa naturaleza, para poder acceder posteriormente a estos a través de los intérpretes adecuados.

Fotograma: Es cada imagen de una película, es decir, un fotograma. La sucesión de fotogramas a gran velocidad produce el efecto de movimiento.

Video digital: Una señal de video representado por computadora y puede leer números binarios que describen un conjunto finito de colores y niveles de luminosidad.