



INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

ESCOM

Trabajo de Titulación

***“Software de Procesamiento de Imágenes en Videos que
Ocupan la Técnica Stop Motion”***

*Que para cumplir con la opción de titulación curricular
en la carrera de*

“Ingeniería en Sistemas Computacionales”

Presentan

**Ayala Castro Salvador
Vázquez Nieves Daniel Augusto
Velázquez Téllez Obed Jonathan**

Directores

**M. en C. Ukranio Coronilla Contreras
M. en C. Yasmín Ivette Jiménez Galán**



ESCOM
Escuela Superior de Cómputo



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO



Registro de titulación: ISCCR045-2011-0032/2015

Fecha: 17 Julio de 2015

Documento técnico

**“Software de Procesamiento de Imágenes en Videos que
Ocupan la Técnica Stop Motion”**

Presentan

Salvador Ayala Castro¹
Vázquez Nieves Daniel Augusto²
Velázquez Téllez Obed Jonathan³

Directores

M. en C. Ukranio Coronilla Contreras
M. en C. Yasmín Ivette Jiménez Galán

RESUMEN

En el presente documento se describe de manera general en lo que se refiere al Análisis, Diseño, Desarrollo e Implementación del “*Software de Procesamiento de Imágenes en Videos que Ocupan la Técnica Stop Motion*”, cuyo objetivo es apoyar en la producción de videos creados con la técnica Stop Motion calculando el movimiento de los objetos en una escena, a partir de un fotograma inicial y la trayectoria definida por el usuario.

Palabras clave – Animación, Fotogramas, Tratamiento Digital de Imágenes.

1 rettersnape@gmail.com

2 megadanx4@msn.com

3 dragon_b_ojazul@hotmail.com

Advertencia

“Este informe contiene información desarrollada por la Escuela Superior de Cómputo del Instituto Politécnico Nacional a partir de datos y documentos con derecho de propiedad y por lo tanto su uso queda restringido a las aplicaciones que explícitamente se convengan.” La aplicación no convenida exime a la escuela su responsabilidad técnica y da lugar a las consecuencias legales que para tal efecto se determinen. Información adicional sobre este reporte técnico podrá obtenerse en: En La Subdirección Académica de la Escuela Superior de Cómputo del Instituto Politécnico Nacional, situada en Av. Juan de Dios Bátiz s/n, Teléfono: 57296000 Extensión 52000

Agradecimientos

Este proyecto ha sido realizado gracias al apoyo de muchas personas. Por eso, nuestro más sincero agradecimiento a **nuestros padres, familiares y amigos** que nos apoyaron incondicionalmente en todo momento.

Queremos dar un agradecimiento especial al **Lic. Eduardo Campos** y su equipo **Lobos Films** por brindarnos su tiempo y la información necesaria para el diseño del sistema, y a la **Escuela Superior de Cómputo** por suministrarnos una educación superior de calidad como Ingenieros en Sistemas Computacionales.

Un especial reconocimiento a nuestros directores:

- **M. en C. Ukranio Coronilla Contreras.**
- **M. en C. Yasmín Ivette Jiménez Galán**

A nuestro jurado evaluador

- **Dr. José Félix Serrano Talamantes**
- **Dr. Benjamín Luna Benoso**
- **M. en C. Luz María Sánchez García**

A los profesores

- **M. en C. María Elena Cruz Meza**
- **M. en C. Gilberto Sánchez Quintanilla**
- **M. en C. Miguel Olvera Aldana**

A nuestro equipo de trabajo por su arduo esfuerzo y su colaboración incluyendo buenos y malos momentos

A todos ellos gracias por las palabras de aliento, consejos, helados, experiencias y motivación.

I. Índice

Advertencia	3
Agradecimientos	4
I. Índice	5
II. Índice de figuras	7
III. Índice de Tablas.....	8
1 Introducción.....	9
1.1 Resumen	10
1.2 Marco teórico	10
1.2.1 Stop Motion.....	10
1.2.2 Física	13
2 Problemática.....	17
2.1 Planteamiento del problema.....	17
2.2 Justificación.....	17
2.3 Estado del Arte.....	17
2.4 Objetivos	18
2.4.1 Objetivo general	18
2.4.2 Objetivos Específicos.....	18
3 Análisis.....	19
3.1 Plan de entregas.....	19
3.2 Análisis de Requerimientos.....	19
3.2.1 Historias de usuario.....	19
3.2.2 Análisis de Requerimientos de usuario	21
3.3 Análisis de Riesgos	22
3.4 Análisis de costo.....	23
3.5 Herramientas de Desarrollo.....	24
3.5.1 Plataforma a utilizar	24
3.5.2 Lenguaje de Programación.....	25
4 Diseño.....	27
4.1 Diagrama de Contexto.....	28
4.2 Diagrama Nivel 1	28
4.3 Diagrama de casos de uso	28
5 Desarrollo	32

5.1	Pantallas del Sistema.....	32
5.1.1	Barra de menús.....	33
5.1.2	Área de visualización de cámara.....	38
5.1.3	Panel de acciones.....	38
5.1.4	Panel de visualización de capturas (Línea de tiempo).....	40
5.2	Implementación.....	41
5.2.1	Desarrollo del componente ListPanel.....	41
5.2.2	Pre visualización.....	41
5.2.3	Cálculos para Área Segura.....	42
5.2.4	Cálculos para la Física.....	42
5.2.5	Estructura del proyecto.....	48
5.2.6	Visualización de la cámara.....	50
5.2.7	Captura de fotogramas.....	52
5.2.8	Control de iluminación.....	52
5.2.9	Previsualización de la animación.....	53
6	Pruebas.....	54
6.1	Pruebas al sistema.....	54
6.2	Pruebas con los usuarios.....	60
7	Conclusiones.....	62
8	Trabajo a futuro.....	63
9	Referencias.....	64
10	Glosario.....	65

II. Índice de figuras

Figura 1.1 Diferentes Relaciones de Aspecto.....	11
Figura 1.2 Área segura en diferentes proporciones	12
Figura 1.3 Ejemplo del efecto papel cebolla	12
Figura 1.4 MRU.....	13
Figura 1.5 MRUA.....	14
Figura 1.6 MCU.....	14
Figura 1.7 Distancia entre puntos MCU.....	15
Figura 1.8 MCUA.....	16
Figura 4.1 Diagrama de Contexto	29
Figura 4.2 Diagrama de nivel 1	30
Figura 4.3 Diagrama de casos de uso	31
Figura 5.1 Pantalla principal.....	32
Figura 5.2 Ventana para crear nuevo proyecto.....	34
Figura 5.3 Abrir proyecto	34
Figura 5.4 Ejemplo de una prueba con papel cebolla.....	35
Figura 5.5 Marco de área segura	36
Figura 5.6 Selector de color.....	36
Figura 5.7 Menú Proyecto	37
Figura 5.8 Nueva Escena.....	37
Figura 5.9 Menú Puntos	37
Figura 5.10 Opción de cambiar colores.....	37
Figura 5.11 Acerca de.....	38
Figura 5.12 Preview Window.....	42
Figura 5.13 Ángulos entre rectas según el cuadrante	46
Figura 5.14 Ángulos desde el eje.....	47
Figura 6.1 MCUA utilizando DragonFrame.....	54
Figura 6.2 MCUA con nuestro sistema	55
Figura 6.3 MCU con DragonFrame.....	55
Figura 6.4 MCU con nuestro sistema	56
Figura 6.5 MRUA con Dragon Frame.....	56
Figura 6.6 MRUA con nuestro sistema	57
Figura 6.7 MRU con Dragon Frame.....	57
Figura 6.8 MRU con nuestro sistema	58
Figura 6.9 Comparativa de tiempos para realizar las pruebas	59
Figura 6.10 Prueba del sistema en un estudio de animación	60
Figura 6.11 Uso de puntos para animar una patineta con velocidad constante	61

III. Índice de Tablas

Tabla 1.1 Ventajas de Stop Motion	9
Tabla 1.2 Desventajas de Stop Motion.....	10
Tabla 2.1 Funcionalidad de las herramientas para realizar Stop Motion	17
Tabla 2.2: Compatibilidad y costos de las herramientas para realizar Stop Motion	18
Tabla 3.1 Historia de usuario 1.....	19
Tabla 3.2 Historia de usuario 2.....	19
Tabla 3.3 Historia de usuario 3.....	20
Tabla 3.4 Historia de usuario 4.....	20
Tabla 3.5 Historia de usuario 5.....	20
Tabla 3.6 Historia de usuario 6.....	20
Tabla 3.7 Historia de usuario 7.....	21
Tabla 3.8 Historia de usuario 8.....	21
Tabla 3.9 Análisis de Riesgos	22
Tabla 3.10 Estadísticas de la W3Schools	24
Tabla 3.11 Estadísticas de la MBA	24
Tabla 3.12 Características de los lenguajes de programación	25
Tabla 3.13 Ventajas y Desventajas de los lenguajes de programación	26
Tabla 5.1 Estructura del archivo XML.....	49
Tabla 6.1 Resultados	58

1 Introducción

Desde tiempos remotos, el ser humano ha sentido la necesidad de expresar sus emociones, ideas y sentimientos. Las artes como la música, el teatro y la danza han satisfecho estas necesidades durante siglos, sin embargo, con el avance de la tecnología, nuevas artes y técnicas han surgido, como la fotografía y el cine, y con ellas nuevas maneras para expresarse.

Entre estas técnicas, destaca el Stop Motion, que es una mezcla de la fotografía y el cine, la cual consiste en capturar, con una cámara fotográfica, imágenes sucesivas que simulen movimiento en objetos inertes.

El Stop Motion es considerado más una técnica de filmografía que de fotografía [1], por lo cual cabe desatacar que tiene ventajas y desventajas en comparación con las otras técnicas de Filmografía.

Las ventajas de la técnica de animación Stop Motion se muestran en la Tabla 1.1 y las desventajas se muestran en la Tabla 1.2.

Tabla 1.1 Ventajas de Stop Motion

Ventaja	Descripción
Realismo	Al poder utilizar prácticamente cualquier objeto para realizar estas animaciones, se da una impresión de que las figuras tienen vida con texturas y volúmenes realistas.
Materiales económicos	Se puede realizar una animación stop Motion básica con una cámara fotográfica y cualquier material disponible.
Posibilidades ilimitadas	La libertad que brinda el stop Motion de utilizar cualquier cosa hace que la imaginación de los animadores sea el único límite.
Versatilidad	Se puede mezclar con las otras técnicas de animación (como animación por computadora) y se acopla fácilmente en producciones de video con actores reales.
Post-producción	Posibilidad de editar los fotogramas posteriormente a su captura para añadir digitalmente cualquier clase de efectos especiales en la animación final.
Exageración de movimientos	Un concepto simple, que al llevar los movimientos al límite, logra que el público inmediatamente centre su atención en el objeto principal.

Fuente: Elaboración propia

Tabla 1.2 Desventajas de Stop Motion

Desventaja	Descripción
Tiempos largos de producción	Debido al esfuerzo que requiere realizar las capturas de cada uno de los fotogramas, además de la creación de modelos y figuras que representan personajes y escenarios, los tiempos de producción de una animación Stop Motion son bastante largos comparados con otras técnicas clásicas.
Alta complejidad	Se deben capturar muchos frames, teniendo cuidado de que los movimientos captados no sean muy abruptos ni exagerados. Es necesario conocer cuánto tiempo tarda un evento desde que inicia hasta que termina. Se debe controlar la iluminación para evitar cambios evidentes en el resultado final, así como condiciones climatológicas dentro de la escena.
Las animaciones de calidad resultan costosas	Se requieren de una gran inversión al usar materiales de calidad con muchos detalles, se necesita equipo fotográfico especializado, software dedicado y un estudio de grabación con condiciones controladas de iluminación.
Costo elevado de las herramientas de software	Las herramientas de software existentes tienen un costo bastante elevado para el usuario común aficionado a realizar animaciones.

Fuente: Elaboración propia

1.1 Resumen

Lo que se pretende con este proyecto es subsanar un poco las desventajas del Stop Motion para facilitar a los usuarios que no tienen conocimientos previos de cómo usar esta técnica, ayudándolos a que sus producciones tengan más consistencia en los movimientos sin necesidad de que piensen demasiado en cómo mover el objeto a animar. Además de proporcionarles herramientas para la captura de los fotogramas necesarios para crear sus animaciones.

1.2 Marco teórico

1.2.1 Stop Motion

El Stop Motion, es una técnica de animación que consiste en aparentar el movimiento de objetos estáticos, como muñecos y marionetas, por medio de una serie de imágenes fijas sucesivas [1] [2] [3] [4]. Para entender mejor esta técnica y cómo debe realizarse es necesario conceptualizar los siguientes aspectos técnicos.

1.2.1.1 Animación

La animación es el proceso de dar vida, interés, espíritu, actividad o movimiento a imágenes, objetos o dibujos estáticos [3].

1.2.1.2 Fotograma

Un fotograma es una sola imagen capturada que forma parte de una animación. También se le conoce como *frame*, *toma* o *cuadro* [3].

1.2.1.3 Antecedentes

La técnica Stop Motion existe desde el siglo pasado, se utilizaba para filmes de formatos cortos o efectos especiales. El pionero de esta técnica fue James Stuart Blackton quien creó el cortometraje llamado “Fases humorísticas de las caras divertidas” en la cual daba vida a unos dibujos hechos en un pizarrón. Entre los primeros experimentos con objetos en movimiento se encuentran “Bewitched Matches” (1913) y “The Automatic Moving Company” (1912). Esta técnica ha sido trascendental en la historia del cine por lo que muchos cineastas han decidido ocuparla y conservarla hasta el día de hoy [4].

1.2.1.4 Relación de aspecto (*aspect ratio*)

Con la finalidad de que el usuario pueda crear una película que se adapte a cierto tipo de pantalla, es necesario considerar la relación de aspecto que es la proporción de una imagen entre su anchura y altura [5], la cual se calcula dividiendo el ancho por la altura de la imagen. Se suele expresar utilizando la notación $\langle x:y \rangle$.

En la Figura 1.1 se muestran las relaciones de aspecto más comunes. Entre ellas se encuentran la 1.78:1, mejor conocida como 16:9 y la 1.33:1 (4:3) que son las relaciones de aspecto más utilizadas en las pantallas de televisión y computadoras; y las relaciones 1.85:1 y 2.39:1 son más utilizadas en el cine [6] [7].

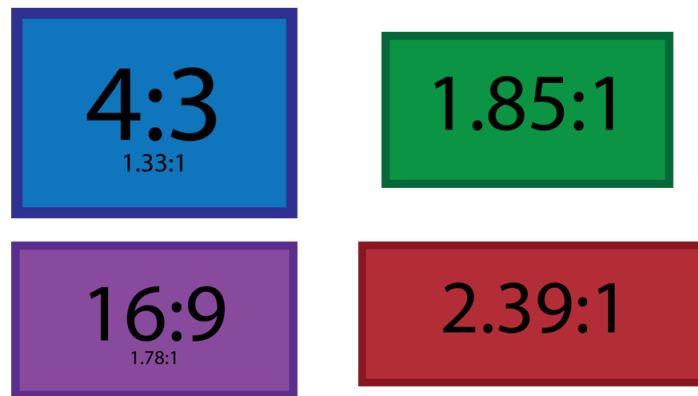


Figura 1.1 Diferentes Relaciones de Aspecto

1.2.1.5 Área segura (*safe frame*)

Además de la relación de aspecto, el usuario necesitará una referencia que le indique cuál es el centro y los límites de la pantalla. Esta referencia delimitará el área segura con líneas guía que se utilizan para garantizar que todos los elementos importantes de una escena sean visibles al mostrarlos en pantallas con diferentes relaciones de aspecto [8]. Una regla fundamental es mantener todos los elementos importantes dentro del 90% del área segura, y

tener en cuenta que después del 95% de ésta área, la imagen podría perderse en algunos televisores.

En la Figura 1.2 se muestran los marcos en diferentes relaciones de aspecto que definen el área segura con un marco amarillo el 95% y el límite dentro del 90% del área segura con un marco blanco.



Figura 1.2 Área segura en diferentes proporciones

1.2.1.6 Papel cebolla (onion skin)

El efecto denominado papel cebolla consiste en visualizar varias imágenes una sobre otra a la vez, superponiendo dos o más imágenes traslúcidas. La Figura 1.3 muestra un ejemplo de cómo se ve este efecto sobre una imagen.

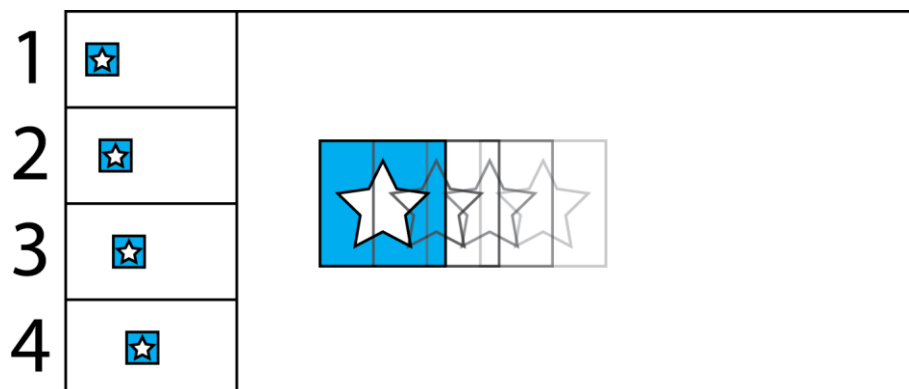


Figura 1.3 Ejemplo del efecto papel cebolla

1.2.2 Física

Con el propósito de facilitar al usuario dónde va a colocar el objeto que desea animar, se calcularán los puntos que describirán la trayectoria de dicho objeto. En cada captura, la trayectoria resultante será resaltada mediante puntos que representarán las posturas o los lugares donde se tendrá que colocar el objeto.

Como se trata de animar objetos diversos, el cálculo de las trayectorias estará limitado a los movimientos rectilíneos y circulares, con y sin aceleración constante [9], los cuales serán explicados a continuación.

1.2.2.1 Movimiento rectilíneo uniforme

Un movimiento es rectilíneo cuando el móvil describe una trayectoria x sobre una recta de la forma $y = mx + b$ donde “ m ” es la pendiente de la recta y “ b ” es la posición de la recta, y es uniforme cuando su velocidad “ v ” es constante en el tiempo “ t ”, dado que su aceleración es nula, de tal forma que un movimiento rectilíneo uniforme está dado por la ecuación:

$$x(t) = A + vt$$

Donde A es el punto de inicio del movimiento o donde cruza la recta el eje de x como se muestra en el siguiente gráfico.

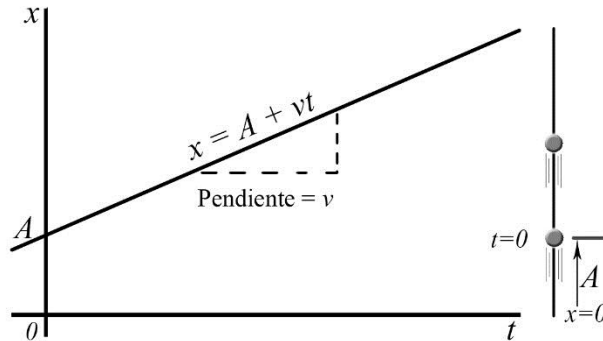


Figura 1.4 MRU

Nos referimos a este tipo de movimiento mediante el acrónimo MRU.

1.2.2.2 Movimiento rectilíneo uniformemente acelerado

El movimiento rectilíneo uniformemente acelerado (MRUA), también conocido como movimiento rectilíneo uniformemente variado (MRUV), es aquel en el que un móvil se desplaza sobre una trayectoria recta (de la forma $y = mx + b$) $x(t)$ estando sometido a una aceleración constante de tal forma que un movimiento rectilíneo uniformemente acelerado está dado por la ecuación:

$$x(t) = A + v_0t + at^2$$

Donde “ A ” es el punto de inicio del movimiento, “ v_0 ” la velocidad inicial del movimiento, “ t ” el tiempo de duración del movimiento y “ a ” la aceleración del movimiento.

De esta forma tenemos la siguiente gráfica la cual nos muestra la variación del movimiento con respecto del tiempo:

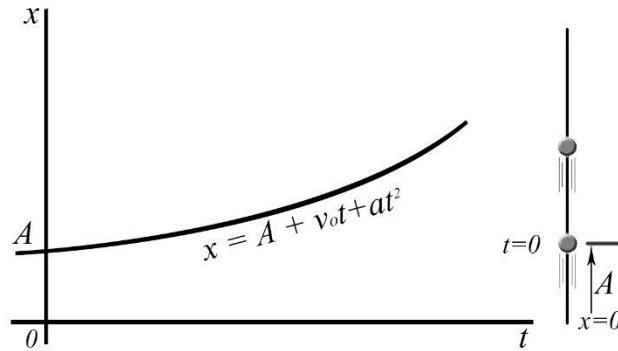


Figura 1.5 MRUA

Nos referimos a este tipo de movimiento mediante el acrónimo MRUA.

1.2.2.3 Movimiento circular uniforme

El movimiento circular uniforme describe un movimiento con una aceleración constante tanto en magnitud como en dirección; también la velocidad es constante, sin embargo, la dirección de la aceleración y la velocidad cambia conforme se mueve el proyectil. La dirección de la aceleración siempre estará apuntando hacia el centro del movimiento mientras que la velocidad será perpendicular a la aceleración, como se muestra en el siguiente gráfico.

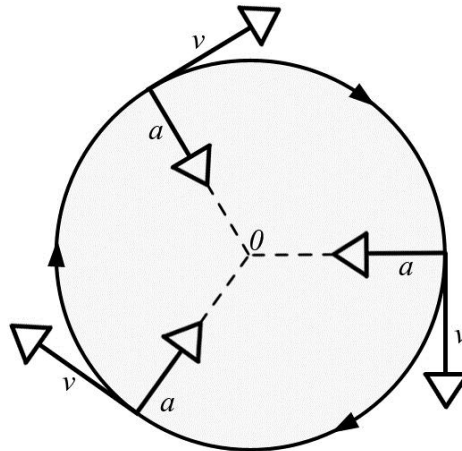


Figura 1.6 MCU

Dadas estas condiciones para un movimiento de un punto P1 a P2 con centro en R y distancia r entre el centro y cualquiera de los puntos en la circunferencia tenemos la longitud de la trayectoria recorrida “e” con un ángulo “ θ ” (establecido en radianes) entre las dos rectas que pasan por el P1, P2 y R respectivamente a una velocidad “v” como se muestra en el siguiente gráfico.

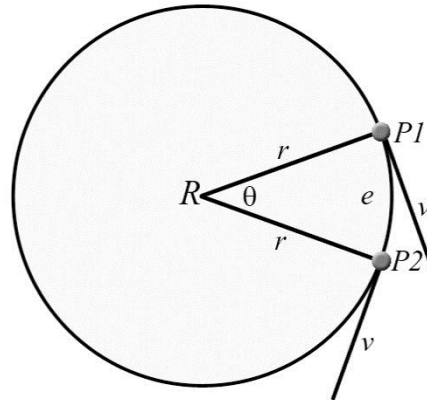


Figura 1.7 Distancia entre puntos MCU

La ecuación de la distancia recorrida entre los puntos es la siguiente:

$$e = r\theta$$

La ecuación de la velocidad en la fracción del tiempo es:

$$v = \frac{e}{\Delta t}$$

Nos referimos a este tipo de movimiento mediante el acrónimo MCU.

1.2.2.4 Movimiento circular uniformemente acelerado

El movimiento circular uniformemente acelerado, MCUA, es un caso particular de la velocidad ya que la magnitud de ésta varía conforme se mueve en la circunferencia y la dirección de la aceleración se mantiene hacia el centro, sin embargo, obtenemos un nuevo tipo de aceleración llamada aceleración tangencial a la cual aumenta con forme aumenta la aceleración que para este caso es uniforme. El gráfico siguiente muestra los diferentes elementos que componen el MCUA.

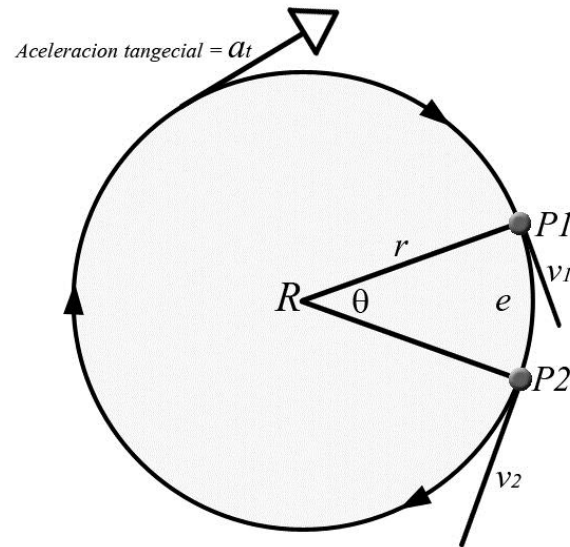


Figura 1.8 MCUA

Las ecuaciones para obtener: la distancia recorrida entre puntos y la de la velocidad son las mismas que en el MCU.

La ecuación de la aceleración en fracción del tiempo a velocidad variable es:

$$a = \frac{dv}{dt}$$

Nos referimos a este tipo de movimiento mediante el acrónimo MCUA.

2 Problemática

2.1 Planteamiento del problema

En los últimos años, la técnica de animación Stop Motion se ha convertido en una estrategia importante para la realización de producciones audiovisuales: películas, cortometrajes, publicidad tanto profesionales como las de aficionados. Sin embargo, esta técnica requiere de equipo y material especializado [2]; pero sobre todo poner un gran cuidado en los detalles para poder brindar al público una animación de calidad. En este sentido, las herramientas de software proveen a las producciones de una gran ayuda para elevar la calidad de las mismas.

Dentro de este contexto, las herramientas de software actuales para la producción de este tipo de animaciones son muy costosas, lo que dificulta que la mayoría de las personas que no se dedican profesionalmente a realizar cortometrajes adquieran dichas herramientas, resultando en animaciones de poca calidad y, por lo tanto, poco agradables para el público.

2.2 Justificación

Debido a que la producción de videos caseros, cortometrajes y largometrajes que emplean la técnica Stop Motion resulta costosa en cuanto a recursos materiales y tiempo, se propone el desarrollo de una herramienta capaz de compensar estos aspectos permitiendo utilizar una cámara web común para minimizar costos; facilitando el trabajo de captura de fotogramas al realizarlo directamente con el software, no sobre el hardware; calculando los puntos de referencia para realizar movimientos precisos de los objetos y mostrando marcos de referencia para una mejor visualización de la animación.

De esta manera, pretendemos desarrollar un sistema capaz de apoyar a los artistas de animación para crear trabajos más realistas invirtiendo menos recursos y tiempo.

2.3 Estado del Arte

En la actualidad existe una gran cantidad de sistemas capaces de auxiliar a las personas que se dedican a crear animaciones de todo tipo; entre ellas, las animaciones que utilizan la técnica Stop Motion. Dentro de estos sistemas, se realizó una comparación de funcionalidad (Tabla 2.1), compatibilidad y costo (Tabla 2.2) de los más sobresalientes.

Tabla 2.1 Funcionalidad de las herramientas para realizar Stop Motion

Software	Captura con webcam o cámara fija	Efecto “Papel Cebolla”	Iluminación automatizada	Captura varios fotogramas a la vez	Idioma
Dragon Stop Motion	Si	Si	Si	No	Inglés
Stopmotionpro	Si	Si	No	Si	Inglés
dflick v1.0	Si	No	Si	No	Inglés
Pinnacle Studio	Si	No	No	No	Español

Fuente: Elaboración propia

Tabla 2.2: Compatibilidad y costos de las herramientas para realizar Stop Motion

Software	Windows	Linux	Mac	Costo
Dragon Stop Motion	Si	No	Si	\$275 USD
Stopmotionpro	Si	No	No	\$185 USD
dflick v1.0	No	Si	No	Software Libre
Pinnacle Studio	Si	No	Si	€69.95

Fuente: Elaboración propia

Como se puede observar, las herramientas que ofrecen una mayor funcionalidad, tienen un precio de venta muy elevado, de acuerdo a las posibilidades económicas de un usuario “no profesional” por lo que éstas quedan fuera de su alcance, limitando su uso a empresas dedicadas a realizar este tipo de animaciones.

Por otro lado, las herramientas gratuitas ofrecen muy poca funcionalidad y ninguna de las herramientas analizadas es capaz de calcular los puntos que seguirá la trayectoria de movimiento del modelo lo que contribuiría a dar una mayor calidad a las producciones.

Otro punto a destacar es que la mayoría de ellas sólo está disponible en inglés y esto representa otro obstáculo para su uso. Con base en lo anterior, se propone desarrollar un sistema que ayude a la producción de videos creados con la técnica Stop Motion.

2.4 Objetivos

Se han planteado los siguientes objetivos que nuestro sistema cumplirá cuando esté finalizado.

2.4.1 Objetivo general

Desarrollar un sistema que ayude a la producción de videos creados con la técnica Stop Motion, capaz de calcular los movimientos subsecuentes a partir de un fotograma inicial y la trayectoria definida por el usuario, ya sea MRU, MRUA, MCU ó MUA.

2.4.2 Objetivos Específicos

- Implementar un algoritmo que nos permita calcular los puntos relevantes para realizar una animación más fluida.
- Hacer el análisis de las trayectorias rectilíneas o curvilíneas que debe recorrer el objeto que se desea animar.
- Con base a un punto de partida y a la trayectoria indicada por el usuario, calcular los puntos donde debe colocarse el objeto a animar.

3 Análisis

3.1 Plan de entregas

Se ha decidido realizar dos entregas a lo largo del desarrollo del proyecto, correspondiendo a las dos etapas de evaluación de las materias Trabajo Terminal I y Trabajo Terminal II

1. Análisis y Diseño
2. Desarrollo y pruebas

La primera entrega corresponde al contenido del presente documento, mientras que la segunda comprenderá el sistema terminado junto con la documentación completa.

3.2 Análisis de Requerimientos

3.2.1 Historias de usuario

Se realizó una entrevista con artistas dedicados a realizar cortometrajes empleando la técnica Stop Motion. Nos comentaron algunas de sus necesidades y dificultades que se les presentan durante la creación de dichas animaciones.

Con base a la información recopilada, se elaboraron las historias de usuario que permitirán realizar un análisis de requerimientos para el sistema más preciso. Estas historias se muestran desde la Tabla 3.1 hasta la Tabla 3.8, donde se muestra la prioridad y el riesgo estimado de cada una de ellas.

Tabla 3.1 Historia de usuario 1

Historia de usuario		
Número: 1	Nombre: Visualizar Cámara	Usuario: Artistas de Stop Motion
Prioridad en el negocio: Alta		Riesgo en el desarrollo: Bajo
Descripción: Necesitamos que el sistema nos muestre las imágenes que capta la cámara en tiempo real.		
Observaciones: Consideramos que este requerimiento será sencillo de implementar.		

Tabla 3.2 Historia de usuario 2

Historia de usuario		
Número: 2	Nombre: Captura de fotogramas	Usuario: Artistas de Stop Motion
Prioridad en el negocio: Alta		Riesgo en el desarrollo: Bajo
Descripción: Queremos que el sistema nos deje hacer capturas de imágenes con la cámara y que se guarden de forma ordenada para que sea fácil hacer la animación final.		
Observaciones: Tampoco habrá problemas con este requerimiento		

Tabla 3.3 Historia de usuario 3

Historia de usuario		
Número: 3	Nombre: Mostrar fotogramas	Usuario: Artistas de Stop Motion
Prioridad en el negocio: Media		Riesgo en el desarrollo: Bajo
Descripción: Nos gustaría que se mostraran los fotogramas ya capturados en la pantalla.		
Observaciones: Será un requerimiento fácil de cumplir.		

Tabla 3.4 Historia de usuario 4

Historia de usuario		
Número: 4	Nombre: Papel Cebolla	Usuario: Artistas de Stop Motion
Prioridad en el negocio: Alta		Riesgo en el desarrollo: Medio
Descripción: Algo importante es que se puedan observar los fotogramas anteriores de forma “encimada” para ir comparando los fotogramas anteriores con la escena actual, con el efecto de papel cebolla.		
Observaciones: Podría complicarse debido a algunas limitaciones en las clases que muestran lo que captura la cámara.		

Tabla 3.5 Historia de usuario 5

Historia de usuario		
Número: 5	Nombre: Vista de la animación	Usuario: Artistas de Stop Motion
Prioridad en el negocio: Baja		Riesgo en el desarrollo: Bajo
Descripción: Sería útil que se pudiera mostrar una vista previa de cómo va la animación con los fotogramas que ya se tienen a diferentes velocidades como 30, 24, 15 y 12 Frames por segundo.		
Observaciones: Consideramos que cumplir este requerimiento no será muy complicado, pero tampoco es una funcionalidad muy necesaria, es por ello que se le asignó baja prioridad.		

Tabla 3.6 Historia de usuario 6

Historia de usuario		
Número: 6	Nombre: Área segura	Usuario: Artistas de Stop Motion
Prioridad en el negocio: Media		Riesgo en el desarrollo: Medio
Descripción: Cuando hacemos animaciones para diferentes formatos como televisión o cine, hay que asegurarnos de que las escenas no se salgan del cuadro, por ello sería útil mostrar los marcos del área segura sobre la vista de la cámara.		
Observaciones: Podría complicarse debido a algunas limitaciones en las clases que muestran lo que captura la cámara.		

Tabla 3.7 Historia de usuario 7

Historia de usuario		
Número: 7	Nombre: Control de Iluminación	Usuario: Artistas de Stop Motion
Prioridad en el negocio: Media		Riesgo en el desarrollo: Bajo
Descripción: En ocasiones, los fotogramas que capturamos presentan niveles distintos de iluminación, nos gustaría que el software pudiera compensar eso de algún modo.		
Observaciones: Se planea incluir controles de iluminación en la interfaz de usuario para poder regularla de forma manual.		

Tabla 3.8 Historia de usuario 8

Historia de usuario		
Número: 8	Nombre: Cálculo de puntos	Usuario: Artistas de Stop Motion
Prioridad en el negocio: Alta		Riesgo en el desarrollo: Bajo
Descripción: Si se pudieran calcular y mostrar los puntos que marcan la trayectoria del movimiento de la figura sería de mucha ayuda al realizar animaciones de figuras que llevan cierta aceleración.		
Observaciones: Este requerimiento es importante porque distinguirá nuestro sistema de otros ya existentes. Para cumplirlo, será necesario que el usuario ingrese ciertos parámetros al sistema.		

3.2.2 Análisis de Requerimientos de usuario

En base a las historias de usuario, se determinaron los requerimientos del sistema:

- El sistema deberá ser capaz de capturar una imagen desde la cámara web (Tabla 3.1).
- El usuario podrá definir:
 - Los puntos clave para calcular de la escena.
 - Tipo de movimiento: rectilíneo uniforme (MRU), rectilíneo uniformemente acelerado (MRUA), circular uniforme (MCU) y circular uniformemente acelerado (MCUA).
 - Dirección del movimiento.
 - Aceleración del movimiento.
- El sistema realizará los cálculos necesarios en base a los parámetros definidos por el usuario para calcular los puntos clave (Tabla 3.8).
- Se podrán mostrar los puntos de referencia que se calcularon y los fotogramas anteriores utilizando el efecto de papel cebolla (Tabla 3.4).
- Se deben mostrar los fotogramas ya tomados (Tabla 3.5) y reproducir una vista previa de la animación (Tabla 3.6).
- El sistema mostrará las líneas guía que delimiten el área segura en diferentes relaciones de aspecto (Tabla 3.6).

- El sistema permitirá cambiar el nivel de iluminación de los fotogramas que va a capturar (Tabla 3.7).

3.3 Análisis de Riesgos

El listado de los posibles riesgos que hay que tomar en cuenta durante el desarrollo del trabajo: probabilidad de evento y el efecto en caso de que suceda se muestra en la Tabla 3.9.

La probabilidad de riesgo se valora como muy bajo si su probabilidad es menor a 10%, bajo si la probabilidad está entre 10-25%, moderado si es entre 25-50%, alto si es entre 50-75% o muy alto si su probabilidad es mayor a 75%. Los efectos del riesgo son valorados como catastróficos, serios, tolerables o insignificantes.

Tabla 3.9 Análisis de Riesgos

Riesgo	Tipo de Riesgo	Descripción	Probabilidad	Efecto
No disponibilidad del hardware	Proyecto	El hardware esencial para el proyecto no está disponible por falta de recursos.	Baja	Serio
Enfermedad del personal	Proyecto y producto	El equipo de análisis, diseño y desarrollo enferma en un momento crítico.	Moderada	Serio
Subestimación del proyecto	Proyecto y producto	El tiempo planeado para realizar el proyecto fue subestimado al hacer la planeación.	Baja	Catastrófico
Cambio en los requerimientos	Proyecto y producto	Los requerimientos podrían cambiar durante el desarrollo del proyecto.	Moderada	Tolerable

Consideramos que los riesgos analizados no constituyen una gran amenaza para el desarrollo del proyecto y podremos hacerles frente sin mucha dificultad en caso de que se presenten.

3.4 Análisis de costo

Para el análisis de costo de nuestro sistema se tomaron en cuenta los siguientes costos:

- Variables
 - Mano de obra (\$ 165,000.00).
 - Tres programadores.
 - 100 horas de análisis y diseño (\$ 250.00 /hr.).
 - 200 horas de programación (\$ 150.00 /hr.).
 - Electricidad (\$ 3,600.00)
 - Ocho meses (\$ 450.00 /mes).
- Fijos
 - Depreciación (\$ 5,688.00)
 - Laptop: Alienware M11x-R2
 - Valor real: \$ 16,000.00
 - Valor de salvamento: \$ 7,300.00
 - Laptop: Toshiba Satellite
 - Valor real: \$ 9,000.00
 - Valor de salvamento: \$ 2,500.00
 - Desktop armada
 - Valor real: \$ 15,000.00
 - Valor de salvamento: \$ 7,000.00
 - Netbook: Dell
 - Valor real: \$ 3,100.00
 - Valor de salvamento: \$ 700.00

Valor real total: \$ 43,100.00
Valor de salvamento Total: \$ 17,500.00
Depreciación total a 3 años: \$ 25,600.00
Depreciación anual: \$ 8,533.00
Depreciación al periodo de 8 meses: \$ 5,688.00
 - Transporte (\$ 6,400.00)
 - Costo diario: \$ 40.00

Costo total: \$ 180,688.00

Cabe destacar que no se calculó un precio de venta porque nuestro sistema no será comercializado; así como no se considero el costo de las licencias de software porque utilizamos herramientas libres y no se tomó en cuenta el costo del equipo de cómputo por representar un costo hundido para el proyecto, únicamente se tomó en cuenta la depreciación del equipo en un periodo de 8 meses.

3.5 Herramientas de Desarrollo

3.5.1 Plataforma a utilizar

La mayoría de los equipos de cómputo actuales dirigidos al público en general, se distribuyen con Windows, lo cual lo convierte, según cifras de la W3Schools (Tabla 3.10) y MBA (Tabla 3.11) en el Sistema Operativo más utilizado.

Tabla 3.10 Estadísticas de la W3Schools

2011	Win7	Vista	Win2003	WinXP	Linux	Mac	Móviles
Septiembre	42.2%	5.6%	0.8%	36.2%	5.1%	8.6%	0.9%
Agosto	40.4%	5.9%	0.8%	38.0%	5.2%	8.2%	0.9%
Julio	39.1%	6.3%	0.9%	39.1%	5.3%	7.8%	1.0%
Junio	37.8%	6.7%	0.9%	39.7%	5.2%	8.1%	0.9%
Mayo	36.5%	7.1%	0.9%	40.7%	5.1%	8.3%	0.8%
Abril	35.9%	7.6%	0.9%	40.9%	5.1%	8.3%	0.8%
Marzo	34.1%	7.9%	0.9%	42.9%	5.1%	8.0%	0.7%
Febrero	32.2%	8.3%	1.0%	44.2%	5.1%	8.1%	0.7%
Enero	31.1%	8.6%	1.0%	45.3%	5.0%	7.8%	0.7%

Fuente: http://www.w3schools.com/browsers/browsers_os.asp

Tabla 3.11 Estadísticas de la MBA

SO	Presencia
Windows XP	45,82%
Windows 7	33,21%
Windows Vista	12,55%
OS X	6,36%
Linux	0,79%
Otros	1,26%

Fuente: MBA Online <http://www.mbaonline.com/>

Dado que la familia Windows cuenta con más del 80% de la participación en el mercado; ésta será la plataforma a utilizar en nuestra herramienta de software ya que uno de los propósitos que se persiguen es que puedan utilizarla la mayor cantidad de interesados en la misma

3.5.2 Lenguaje de Programación

Se tomaron en cuenta ciertos aspectos al momento de decidir qué lenguaje de programación se utilizaría para el desarrollo de la herramienta de software [10]. Los más relevantes fueron:

- El tiempo: Debido a las restricciones de tiempo para la entrega del trabajo terminal, no teníamos tiempo suficiente para aprender un nuevo lenguaje de programación, teníamos que emplear alguno ya conocido.
- La metodología empleada: La metodología de Programación Extrema (XP) establece que deben poder realizarse cambios sobre la marcha, por ello requerimos un lenguaje sencillo de fácil aprendizaje, que sea cómodo para trabajar y a la vez intuitivo.
- Las herramientas existentes: Se pensó en los diferentes Entornos de Desarrollo Integrados (IDE) disponibles para cada lenguaje de programación. Algunos hacen la tarea de programación bastante sencilla con editores para la interfaz gráfica y autocompletando el código mientras se va escribiendo.
- Capacidades extras: El lenguaje a utilizar debe contar con bibliotecas o Interfaces de Programación de Aplicaciones (API) especializadas para el tratamiento digital de imágenes y para el funcionamiento de la cámara digital.

En base a estos aspectos, se realizó un análisis de las características (Tabla 3.12) de cada lenguaje de programación candidato para realizar nuestro trabajo terminal. Además se analizaron las ventajas y desventajas (Tabla 3.13) de cada uno de ellos.

Tabla 3.12 Características de los lenguajes de programación

Característica	Java	C++	C#
Orientado a Objetos	Si	Si	Si
Manejo de eventos	No	No	Si
Documentación	Si	Si	Si
Manejo de Cámara	Si	Si	Si
Windows/Mac/Linux	Si	Si	Si
Hilos	Si	Si	Si
API o Bibliotecas para tratamiento de imágenes	Si	Si	Si
IDE	Si	Si	Si

Fuente: Elaboración propia

Tabla 3.13 Ventajas y Desventajas de los lenguajes de programación

Lenguaje	Ventajas	Desventajas
Java	<ul style="list-style-type: none"> • Conocido por todo el grupo de trabajo. • Portabilidad del código. • Existen numerosas API de código abierto completamente gratuitas. • Cuenta con diferentes IDE para desarrollar. • Es orientado a objetos. • El SDK, los IDE y las API son gratuitas y están disponibles en internet. • Dispone de suficiente documentación desde el sitio oficial de Oracle. 	<ul style="list-style-type: none"> • Rendimiento considerablemente inferior en aplicaciones que requieren una respuesta rápida. • Requiere tener la máquina virtual de Java instalada.
C#	<ul style="list-style-type: none"> • Es bastante intuitivo • El rendimiento es generalmente mayor que en Java • Se puede desarrollar utilizando el IDE Visual Studio • Mayor orientación a objetos. • Integración total con los componentes del Framework .NET nativos de Windows • Cuenta con mucha documentación disponible. • Aplicaciones multi-hilo simplificadas. • Facilidad de uso y ambiente de trabajo muy cómodo. 	<ul style="list-style-type: none"> • El IDE Visual Studio requiere mucha memoria y espacio en disco duro. • Se requiere de internet para consultar toda la documentación.
C++	<ul style="list-style-type: none"> • Creación de funciones. • Tratamiento de la memoria más específico • Permite el uso total de apuntadores • Es un lenguaje muy robusto que permite la creación de todo tipo de aplicaciones dependiendo del uso que se le dé al lenguaje. 	<ul style="list-style-type: none"> • No cuenta con un recolector de basura para liberar los recursos. • Los diferentes tipos de herencia hacen que la programación se vuelva compleja y confusa. • No posee un sistema de tipos unificado como Java o C#. • Uso complejo de bibliotecas dinámicas. • No tiene un estándar bien definido.

Fuente: Elaboración propia

Las tablas comparativas anteriores muestran que todos los lenguajes seleccionados tienen capacidades similares. No parece haber una diferencia decisiva que nos lleve a preferir un lenguaje sobre otro basándose en éstas, ya que los tres lenguajes analizados cumplen las funciones requeridas para el trabajo terminal, que es el manejo de la cámara web. Sin embargo, hemos decidido que utilizaremos el lenguaje C# debido a los conocimientos y experiencia que tenemos, además de la facilidad de uso y comodidad al programar que nos brinda el IDE de Visual Studio.

4 Diseño

La metodología XP es menos orientada a los documentos y más orientada al código [11]. Es por esa razón que una vez elaboradas las historias de usuario y ya que se realizó el análisis de requerimientos sólo realizaremos el diagrama de contexto (Figura 4.1) con un nivel de profundidad 1 (Figura 4.2) y el diagrama de casos de uso (Figura 4.3).

Ésta metodología permite realizar cambios en cualquier momento, ya sea en los requerimientos o en los procesos, haciendo estos últimos adaptables.

Una de las manifestaciones de las metodologías ágiles es que es preferible el software que funciona sobre la documentación exhaustiva, y debido a la limitante de tiempo que tenemos para la entrega del sistema, consideramos que emplear esta metodología nos ayudará a cumplir en tiempo y forma con la entrega del trabajo terminal.

Considerando que los documentos son como barricadas, que limitan el desarrollo práctico del proyecto, al tener poca documentación a la cual apegarse, se pueden generar nuevas ideas conforme la marcha, se pueden contemplar nuevas posibilidades y se tiene más libertad para realizar cambios en el futuro.

El desarrollo de nuestro proyecto se va a basar en 4 actividades básicas:

- Codificar
- Hacer pruebas
- Escuchar
- Diseñar

Y utilizaremos algunos artefactos como son:

- Las historias de usuario
- Pruebas de aceptación
- Pruebas de integración
- Plan de entrega
- Código

4.1 Diagrama de Contexto

En este diagrama (Figura 4.1) se muestran las diferentes interacciones de entrada y salida que el sistema contemplará [12], entre las entradas del usuario están la trayectoria que seguirá cada punto y su aceleración, así como los efectos que desee aplicar al panel de captura de imágenes. El sistema recibirá las solicitudes de captura de fotogramas y entregará al usuario los fotogramas que capturó. Además, el sistema mostrará los resultados de los puntos que se calcularon y los efectos de resaltado y papel cebolla.

4.2 Diagrama Nivel 1

El diagrama de la Figura 4.2 de nivel uno muestra con más detalle el proceso que recibirá las entradas para brindarle las salidas deseadas al usuario [12]. Se contemplan 3 subprocesos principales: Calcular puntos clave, que toma los parámetros que ingresa el usuario sobre el movimiento que desea realizar y calcula los puntos clave donde deberá colocar la figura. Visualizar cámara que tomará los resultados del proceso anterior y las imágenes que captura la cámara en tiempo real para mostrar los resultados y control de la cámara que se encargará de capturar las imágenes de la cámara web.

4.3 Diagrama de casos de uso

En este diagrama (Figura 4.3) se han contemplado los requerimientos determinados en las historias de usuario [12]. Se contemplan 3 casos de uso principales en los que actúa el usuario: Capturar Fotograma, Ingresar parámetros y pre visualizar aplicación.

Cada uno de ellos implica otros casos de uso cuyo propósito es cubrir todos los requerimientos contemplados anteriormente.

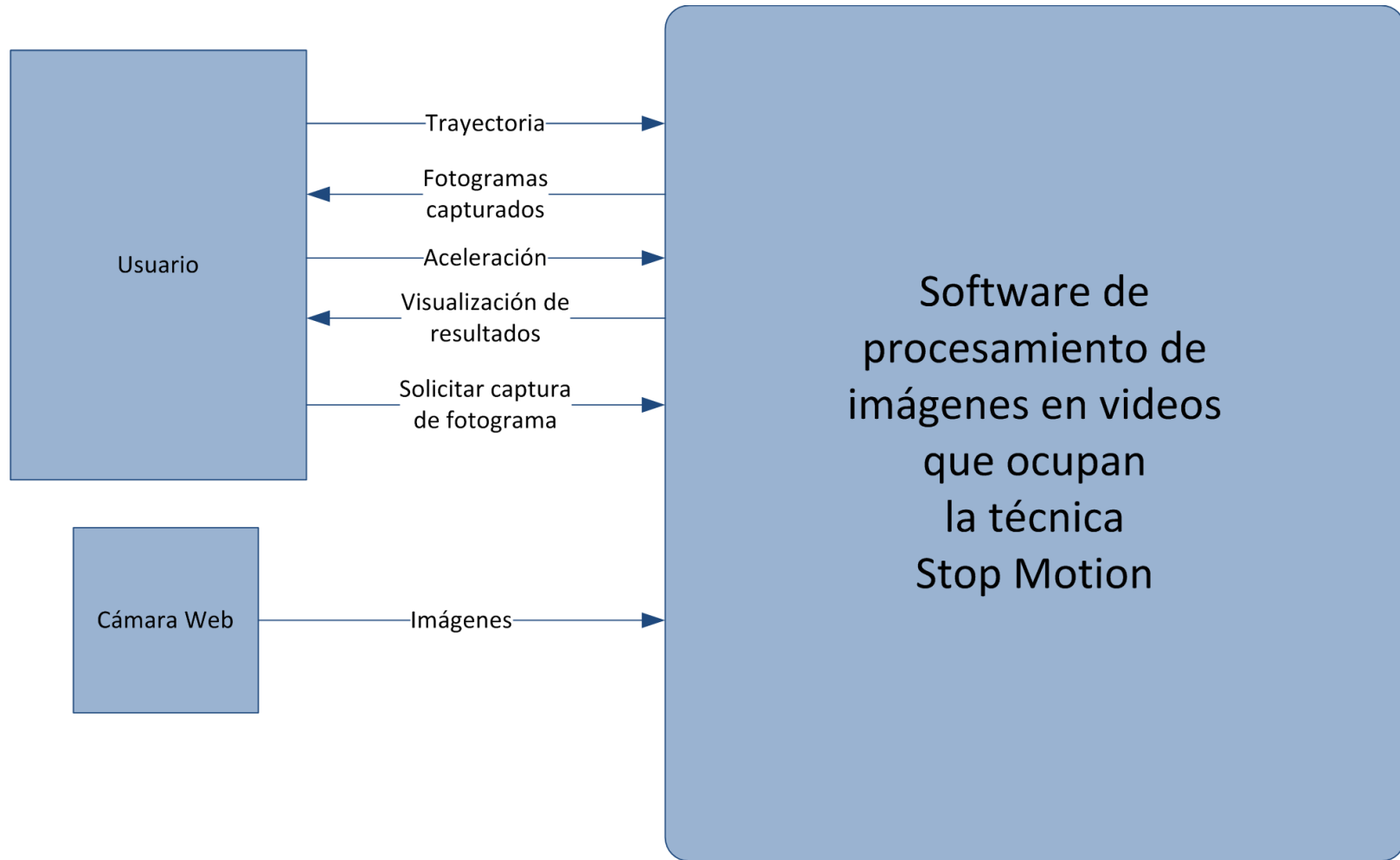


Figura 4.1 Diagrama de Contexto

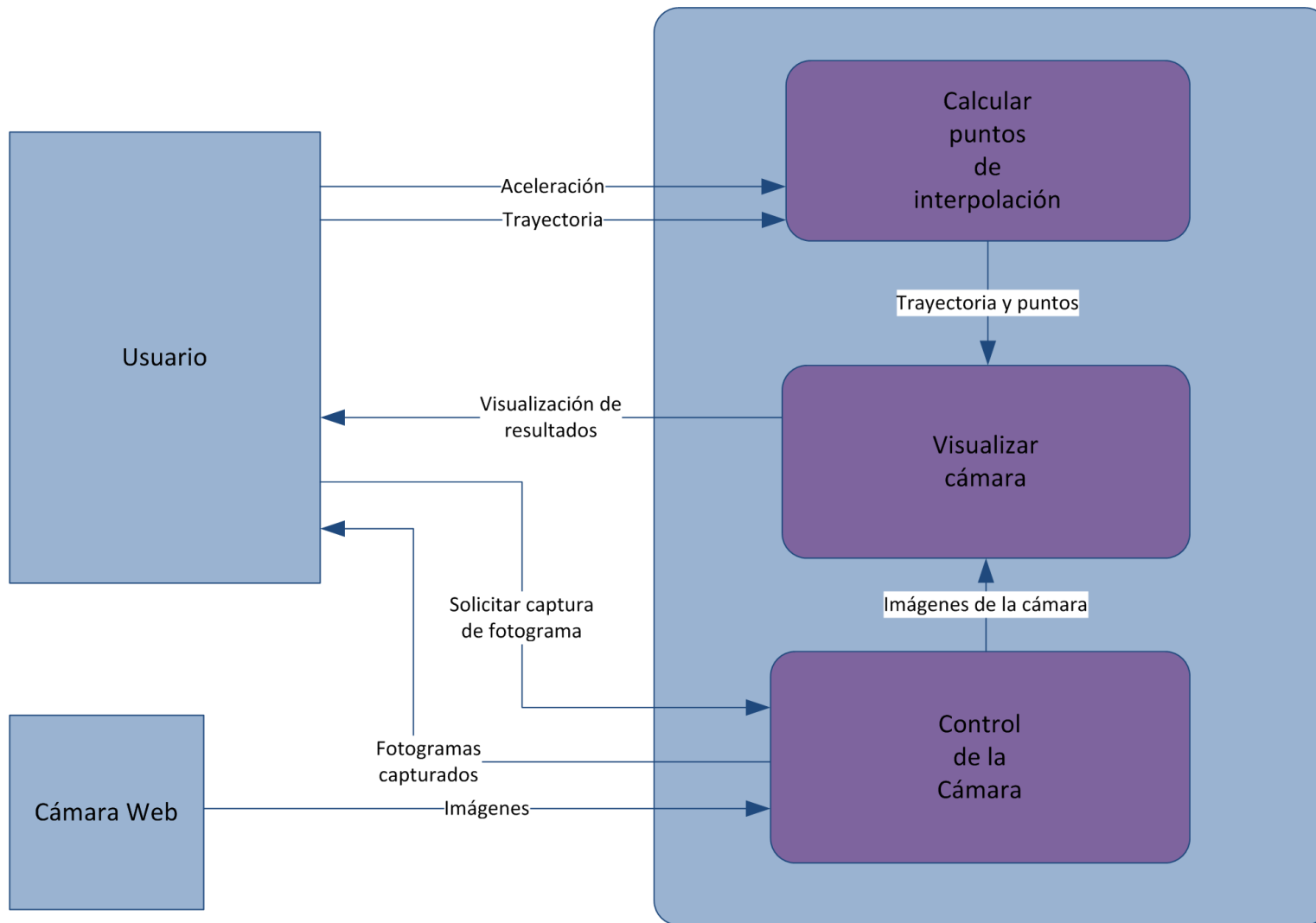


Figura 4.2 Diagrama de nivel 1

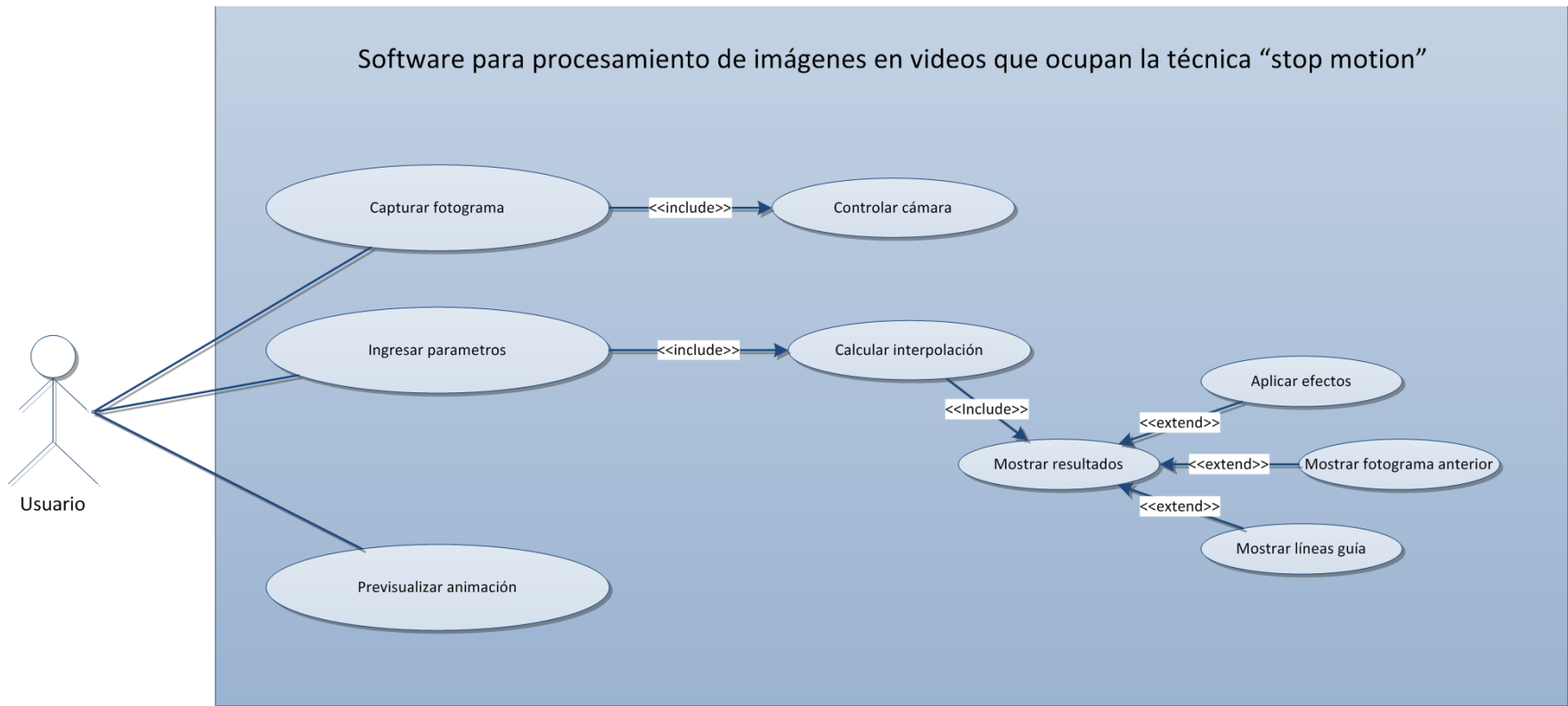


Figura 4.3 Diagrama de casos de uso

5 Desarrollo

El software ofrece un ambiente gráfico al usuario que le facilita su uso, por medio de atajos del teclado (hotkeys) intuitivos, los cuales serán descritos más adelante. También ofrece menús con las mismas funcionalidades que se encuentran en la interfaz principal, con el objetivo de poder prescindir del mouse, utilizando en su lugar combinaciones de teclas y poder acceder a cualquier funcionalidad del software.

5.1 Pantallas del Sistema

La pantalla principal del sistema se divide en 4 partes principales como se puede observar en la Figura 5.1, las cuales son:

1. Barra de menús
2. Área de visualización de cámara
3. Panel de acciones
4. Panel de visualización de capturas (Línea de tiempo)

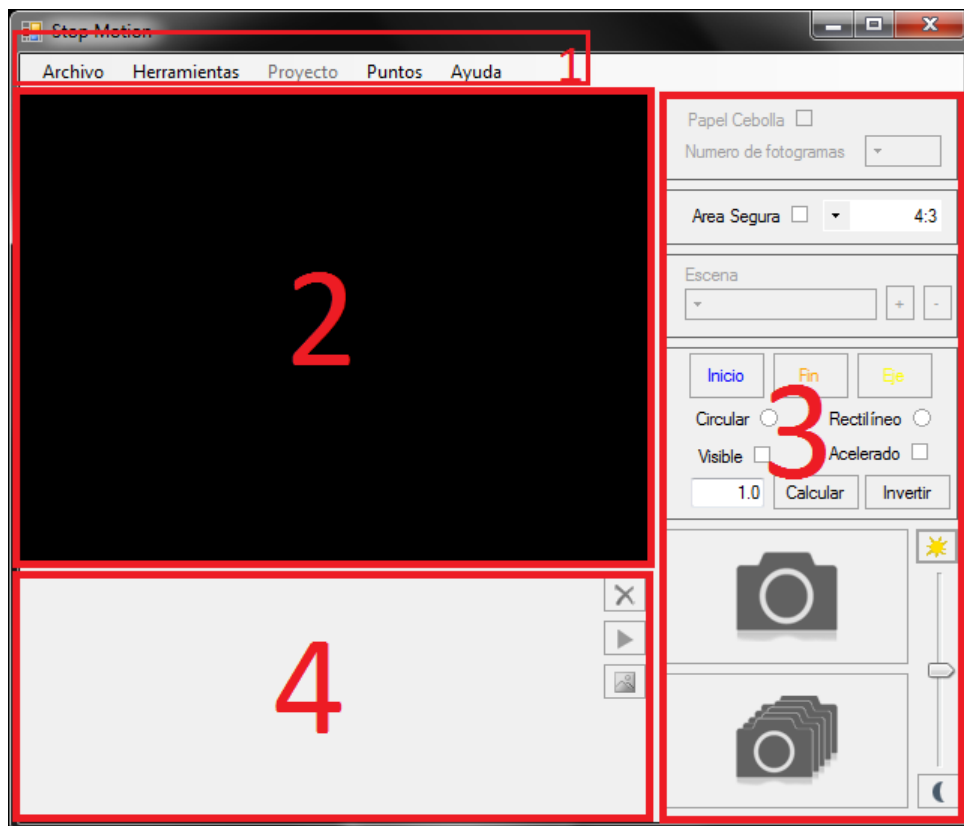


Figura 5.1 Pantalla principal

5.1.1 Barra de menú

En la barra de menú se encuentran las opciones de *Archivo*, *Herramientas*, *Proyecto*, *Puntos* y *Ayuda*. El siguiente árbol es una descripción de todos los menús dentro de la barra:

- Archivo
 - Nuevo Proyecto
 - Abrir
 - Cerrar
- Herramientas
 - Papel Cebolla
 - Habilitar/Deshabilitar
 - 1 fotograma
 - 2 fotogramas
 - 3 fotogramas
 - Área Segura
 - Habilitar/Deshabilitar
 - Televisión (4:3)
 - Panorámico (16:9)
 - PC (16:10)
 - Cine (1.85:1)
 - Cine (2.39:1)
 - Seleccionar color...
 - Iluminación
 - Subir
 - Bajar
- Proyecto
 - Capturar Fotograma
 - Agregar Escena
 - Eliminar Escena
- Puntos
 - Mostrar/Ocultar
 - Inicial
 - Final
 - Eje
 - Cambiar colores
 - Inicial...
 - Final...
 - Eje...
 - Puntos intermedios...
 - Punto siguiente...
- Ayuda
 - Acerca de...

El menú *Archivo* es esencial en la funcionalidad del software ya que desde aquí se puede crear un *Nuevo Proyecto*, *Abrir* uno existente o *Cerrar* el programa.

Al crear un *Nuevo Proyecto*, se despliega una ventana (Figura 5.2) nueva donde se puede escribir el nombre del proyecto y la ruta donde será guardado, así como los fotogramas por segundo.

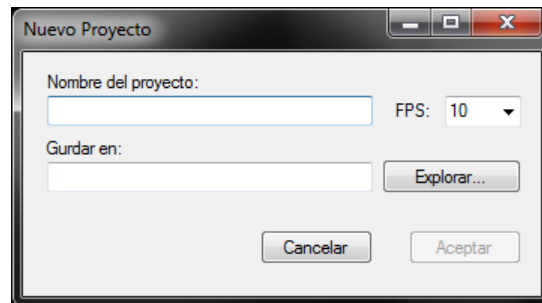


Figura 5.2 Ventana para crear nuevo proyecto

Al *Abrir* un proyecto existente se muestra un diálogo de apertura de archivos (Figura 5.3) que filtrará los archivos con el formato de proyecto que el software soporta.

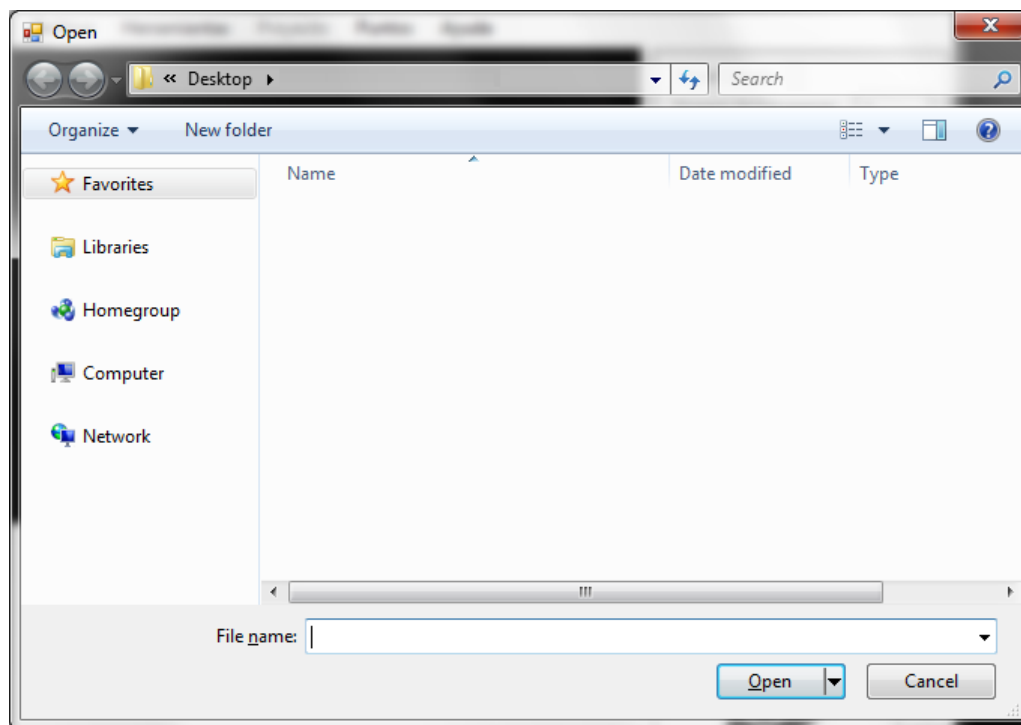


Figura 5.3 Abrir proyecto

Al *Cerrar* el programa no se pierde la información de las escenas ya que ésta se guarda cada vez que hay algún cambio en el proyecto.

El menú *Herramientas* permite al usuario activar o desactivar el Papel Cebolla y Área Segura así como ajustar la iluminación de la toma.

El Papel Cebolla tiene la opción de visualizar hasta 3 fotogramas, siempre y cuando estos existan. En la Figura 5.4 se observa cómo el papel cebolla sirve de referencia para saber qué tanto se ha movido el objeto.

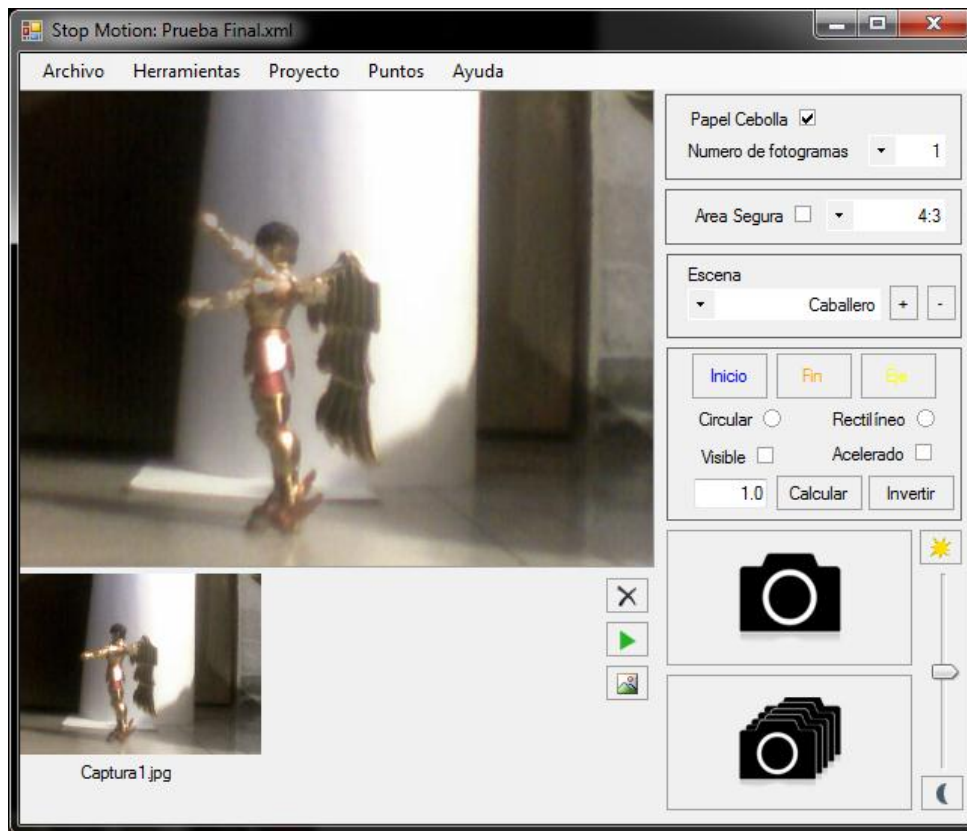


Figura 5.4 Ejemplo de una prueba con papel cebolla

El Área Segura (Ilustrada en la Figura 5.5) permite visualizar los marcos con proporciones 4:3, 16:9, 16:10, 1.85:1 y 2.39:1. Y sirve para que el objeto no se salga de cuadro ni se pierdan detalles de las tomas.



Figura 5.5 Marco de área segura

También permite elegir el color del marco mediante un cuadro de selección de color como el de la Figura 5.6.

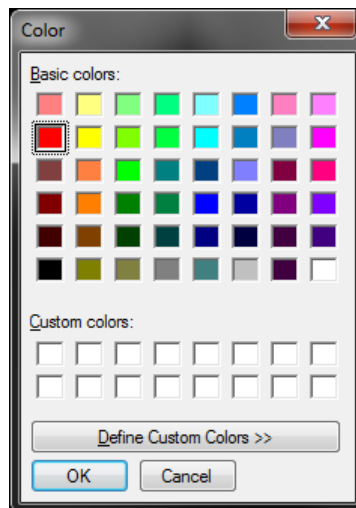


Figura 5.6 Selector de color

El menú Proyecto (Figura 5.7) permite al usuario Agregar Escenas y Capturar Fotogramas (siempre y cuando exista una escena creada y activa).

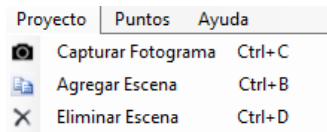


Figura 5.7 Menú Proyecto

Al agregar una escena aparecerá un cuadro de texto (Figura 5.8) donde se colocará el nombre de la nueva escena.

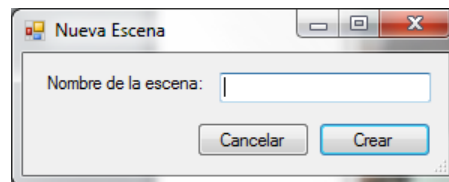


Figura 5.8 Nueva Escena

El menú Puntos (Figura 5.9) permite al usuario manejar el control de los puntos ya que puede activarlos o desactivarlos así como cambiar su color o posición. Los puntos que podrá manejar el usuario serán el punto inicial, punto final, y un punto eje (siempre y cuando existan los puntos inicial y final) para definir un movimiento.

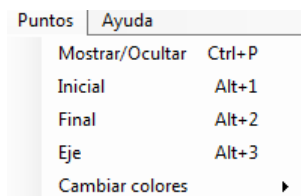


Figura 5.9 Menú Puntos

Los colores de cada punto dibujado se pueden cambiar (ver Figura 5.10). Es posible cambiar el color del punto inicial, del final, de los puntos intermedios y del punto que indica dónde debe colocarse la figura, al cual llamaremos “Punto Siguiente”. Al cambiar los colores aparecerá un diálogo selector de color. Los colores regresarán a los valores predeterminados al iniciar nuevamente el programa.

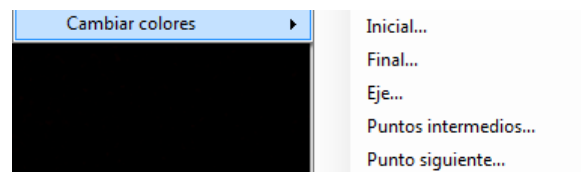


Figura 5.10 Opción de cambiar colores

El menú Ayuda muestra la opción “acerca de” que abre una ventana (Figura 5.11) donde viene el nombre y versión del programa, los alumnos y directores involucrados en la creación de este proyecto.



Figura 5.11 Acerca de

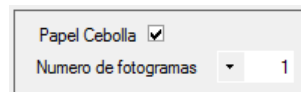
5.1.2 Área de visualización de cámara

En esta área se puede visualizar el objetivo de la cámara, así como los efectos que se apliquen desde el menú Herramientas, o bien, el Panel de Acciones.

5.1.3 Panel de acciones

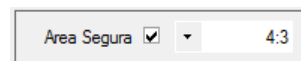
Aquí se encuentran las herramientas principales divididas en seis secciones:

5.1.3.1 Papel Cebolla



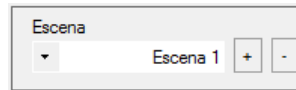
Aquí se puede Habilitar o Deshabilitar el Papel Cebolla por medio del Cuadro de Selección (CheckBox), y elegir el Número de fotogramas a mostrar desde el Cuadro de Lista Desplegable (ComboBox). Esta funcionalidad sólo ésta disponible cuando hay un proyecto abierto y una escena esta activa.

5.1.3.2 Área Segura



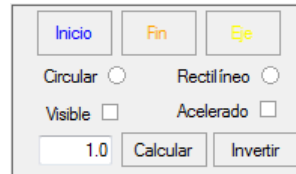
Aquí se puede Habilitar o Deshabilitar el Área Segura por medio del CheckBox, y elegir su Relación de Aspecto desde el ComboBox. Esta funcionalidad está disponible en todo momento.

5.1.3.3 Escena








Aquí se puede seleccionar la escena en la que se desea trabajar mediante el ComboBox. También se puede Agregar (+) o Eliminar (-) una escena con sus respectivos botones. Esta sección sólo está disponible si hay un proyecto abierto.

5.1.3.4 Movimiento



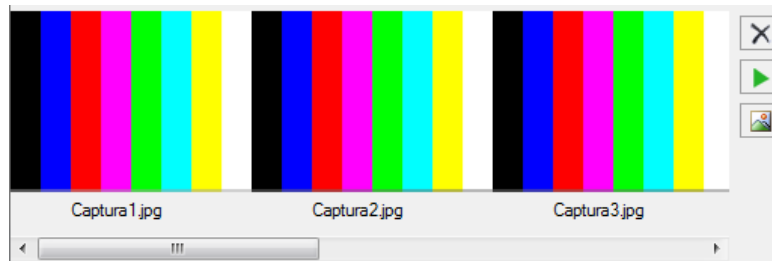
Aquí se puede seleccionar si se quiere colocar o mover un punto inicial (**Inicio**), final (**Fin**) o eje (**Eje**) en el área de captura mediante sus respectivos botones, al dar clic por primera vez se podrá colocar el punto, de ya estar puesto, se podrá modificar su posición. El CheckBox “Visible” (**Visible**) permite habilitar o deshabilitar la visibilidad de los puntos que se han dibujado sobre el área de captura, El Cuadro de texto () indica el tiempo en el que se quiere realizar el movimiento, dos ComboBox para que el usuario decida que tipo de movimiento desea calcular ya sea rectilíneo (**Rectilíneo**) o circular (**Circular**). Los movimientos requieren que existan los puntos inicial y final, adicionalmente el movimiento circular, requiere de un punto eje. El CheckBox “Acelerado” (**Acelerado**) se utiliza para indicar si el movimiento a calcular será acelerado. El botón “Calcular” (**Calcular**) sirve para que el programa calcule el movimiento y lo muestre sobre el área de captura. En caso de que el movimiento muestre una trayectoria en el sentido opuesto al deseado, se cuenta con el botón “invertir” (**Invertir**).

5.1.3.5 Captura

En esta área se pueden tomar las capturas con el botón con una imagen de una cámara fotográfica () , así también se tiene un botón de captura rápida el cual permitirá tomar tantas capturas sean posibles (según los frames por segundo con los que se haya creado el proyecto) hasta ser presionado otra vez dicho botón () , también se tiene un “ TrackBar” () cuya funcionalidad es controlar la iluminación del área de captura, igualmente se cuenta con dos botones que permiten aumentar () o disminuir () la iluminación.


5.1.4 Panel de visualización de capturas (Línea de tiempo)








En este panel podemos encontrar los frames tomados en la escena, mostrados en una línea de tiempo, así como tres botones de acción: eliminar, reproducir vista previa y ampliar.



Eliminar ()

Este botón nos sirve para eliminar el último frame capturado.

Reproducir ()

Al dar clic en este botón aparecerá una nueva ventana con un área de visualización la cual tiene todos los frames de la escena. Para reproducirla hay que dar clic en el botón de reproducir () si se quiere detener se da clic en el botón detener (), si se quiere adelantar o atrasar un frame se tiene los siguientes botones ( ) , en caso de querer adelantar todo o atrasar todo se tienen los siguientes botones ( ) así también tenemos un ListBox para seleccionar la velocidad de reproducción del video (FPS: 24 ) , un CheckBox para decidir si se verá repetidamente el video o sólo una vez (Repetir) y también un “trackbar” o barra de desplazamiento para navegar entre los fotogramas.

Ampliar ()

Finalmente este botón nos sirve para ver el frame actual en una nueva ventana, esto también se puede hacer desde la línea de tiempo dando doble clic en la imagen que se desea agrandar.

5.2 Implementación

En esta sección se describirá con mayor detalle cómo se implementaron los módulos más relevantes o importantes del proyecto

5.2.1 Desarrollo del componente ListPanel

Para poder mostrar al usuario las imágenes pertenecientes a cada escena fue necesaria la creación de un propio componente capaz de almacenar dichas imágenes, y mostrarlas al usuario, tomando en cuenta que almacenar imágenes requiere de mucha memoria al momento de ser utilizadas en el programa. Así fue como se desarrolló la clase ListPanel (ver Figura 5.1, panel 4), la cual cuenta con un área de visualización de imágenes, y en caso de ser necesario, se activa una barra de desplazamiento para poder desplazarse entre las imágenes de la escena. Durante el desarrollo, esta clase sufrió de muchos cambios, ya que como se mencionó anteriormente, almacenar imágenes ocupa mucha memoria, y con unas pocas imágenes el programa registraba una gran cantidad de uso de memoria, llegando a un punto en que era inestable y ocurría una excepción por la falta de memoria. Dado que no era necesario tener las imágenes en su resolución completa mientras eran mostradas en el ListPanel, se optó por almacenar miniaturas (thumbs) de las imágenes en vez de la imagen en su resolución original, que tratándose de una imagen en tamaño VGA ocupa alrededor de 1 MB por imagen, pudiendo éstas llegar a ocupar más de 10 MB a mayores resoluciones. También se consideró que el usuario podría querer ver alguna de las imágenes capturadas, en su resolución original para analizar más a fondo la escena en ese momento. Por lo que se implementó la opción de mostrar individualmente las imágenes en una ventana emergente, mediante la combinación de presionar la tecla de control (ctrl) y click en la imagen deseada.

5.2.2 Pre visualización

Durante todo el proceso de captura en cada escena, es indispensable contar con una herramienta que permita visualizar el trabajo en cualquier momento, con el fin de detectar errores a tiempo en caso de haberlos y corregirlos inmediatamente. Esto nos llevó a crear una clase a la que nombramos PreviewWindow, la cual permite reproducir las imágenes de cualquier escena. Al diseñar esta clase, se tuvo en mente la interfaz de un reproductor de video, por eso se cuenta con controles similares a los que se encuentran en casi todos los reproductores (Figura 5.12). La reproducción se puede hacer a diferentes velocidades de fotogramas por segundo (FPS), esto nos llevó a hacer diversas pruebas donde se observó que algo afectaba el rendimiento, y que la velocidad con que se reproducía la animación se reducía muy drásticamente, dicho problema era el uso de memoria en áreas del programa donde no se requería mientras se reproducía la vista previa, esto se solucionó reduciendo el tamaño de las imágenes en memoria y deteniendo momentáneamente la captura de la cámara. La clase final reproduce la vista previa a la velocidad correcta. Cabe destacar que esta clase no crea un archivo de video que pueda ser reproducido fuera del programa.

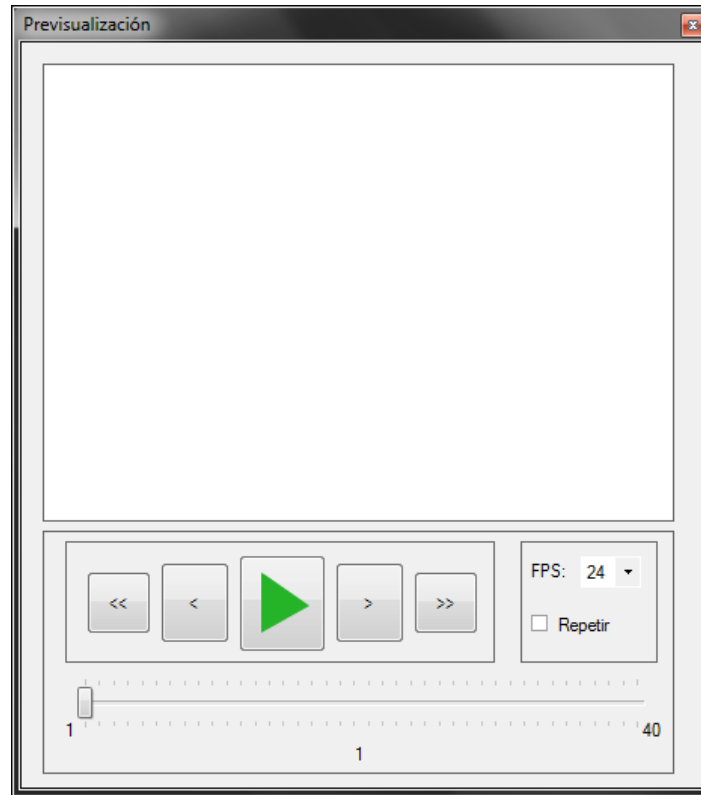


Figura 5.12 Preview Window

5.2.3 Cálculos para Área Segura

Para mostrar el marco seguro en el Área de Visualización de la Cámara inicialmente se hacía utilizando una imagen con las líneas del área segura sobre un fondo transparente, que cumplía con el propósito deseado pero se presentaron problemas cuando la resolución de la cámara era diferente a la de la imagen del Área Segura.

Este inconveniente se solucionó dibujando el área sobre la imagen, en vez de utilizar una imagen prediseñada. El tamaño del área segura se calculó en base al tamaño de la imagen capturada, de la siguiente forma:

Se obtiene el tamaño de la imagen y se calcula el centro de la misma, se crea un rectángulo con un anchura de 90% respecto de la imagen capturada, y el alto se calcula dependiendo de la relación de aspecto, se dibuja sobre la imagen capturada, y para identificar el centro también se dibuja una cruz.

5.2.4 Cálculos para la Física

Con la finalidad de mostrar al usuario los puntos que describan un movimiento que él elija, se necesitó de cálculos físicos y matemáticos los cuales según los puntos recibidos y las opciones seleccionadas mostraría una u otra cosa.

Se realizó una clase Física la cual contendrá todos los métodos para calcular los movimientos y sus necesarias operaciones, esta clase devuelve una lista de objetos del tipo Point la cual contiene todas las coordenadas del movimiento.

5.2.4.1 Movimiento Rectilíneo

5.2.4.1.1 Uniforme

Primeramente se realizó el MRU, donde se tuvo que calcular a partir de los puntos Inicial y Final la distancia entre ellos, obteniendo las coordenadas x y y de cada punto y restándolos entre sí. Con el tiempo establecido por el usuario más la distancia entre puntos se obtuvo la velocidad de la siguiente ecuación:

$$v = \frac{dx}{dt}$$

La cual nos indica que la velocidad es igual a la distancia recorrida en un cierto tiempo, obteniendo la velocidad y debido a que es la misma en todo el movimiento, se obtiene para cada frame el tiempo en el que este aparece (velocidad de frames con la que se creó el proyecto) y se multiplica por la velocidad para saber la distancia recorrida hasta el momento en el que el frame aparece. Debido a que sólo con esto no se podía saber la coordenada donde se encontraba el punto del movimiento se necesitó hacer un sistema de ecuaciones, el cual indicará donde se encontraba esta coordenada, las ecuaciones se muestran a continuación:

Como movimiento tenemos una recta del tipo “ $y = mx + b$ ” tenemos el punto inicial del movimiento sobre esta misma recta y la distancia que se tiene que recorre desde este punto, también sabemos por trigonometría que la distancia entre dos puntos es:

$$d = \sqrt{(x - x_0)^2 + (y - y_0)^2}$$

Donde d es la distancia entre puntos (la distancia avanzada desde el punto inicial), x y y las coordenadas del punto final (para nuestro caso la coordenada desconocida) y x_0 y y_0 las coordenadas del punto inicial.

De esta ecuación elevaremos todo al cuadrado para que desaparezca la raíz quedando de la siguiente forma:

$$d^2 = (x - x_0)^2 + (y - y_0)^2$$

Sabemos también que la pendiente de una recta es $m = \frac{(y-y_0)}{(x-x_0)}$ (donde m es la pendiente) y debido a que tenemos la ecuación de la recta también poseemos el valor de la pendiente.

Multiplicaremos la ecuación de la distancia por lo siguiente $\left(\frac{1}{(x-x_0)}\right)^2$

$$\left(\frac{1}{(x - x_0)}\right)^2 * d^2 = ((x - x_0)^2 + (y - y_0)^2) * \left(\frac{1}{(x - x_0)}\right)^2$$

Y nos queda de la siguiente forma:

$$\left(\frac{d}{(x-x_0)}\right)^2 = \left(\frac{(x-x_0)}{(x-x_0)}\right)^2 + \left(\frac{(y-y_0)}{(x-x_0)}\right)^2$$

Podemos observar que en la parte de la derecha el primer término es uno (ya que la fracción tiene el mismo numerador como denominador) y el segundo es la pendiente al cuadrado (recordando la ecuación de la pendiente) así que reducimos términos y queda de la siguiente forma:

$$\left(\frac{d}{(x-x_0)}\right)^2 = 1 + m^2$$

En esta ecuación observamos que conocemos todos los términos excepto x el cual es la coordenada del eje x que estamos buscando así que despejamos x de la ecuación y nos queda de la siguiente forma:

$$x = x_0 + \sqrt{\left(\frac{d}{(1+m^2)}\right)^2}$$

Teniendo ya la coordenada en x del punto sólo se sustituye en la recta $y = mx + b$ y se obtiene el valor de y . Después de esto se agrega la coordenada a la lista de puntos.

5.2.4.1.2 Uniforme Acelerado

Para el MRUA se ocuparon las mismas fórmulas para obtener la distancia entre puntos, la coordenada del punto después del movimiento y la velocidad, sin embargo esta velocidad obtenida será la velocidad final del movimiento, adicionado a que para obtener la aceleración se utilizó la siguiente fórmula:

$$a = \frac{dv}{dt}$$

La cual nos indica que la aceleración es igual a la variación velocidad en un cierto tiempo, la variación de la velocidad se obtiene de la siguiente manera:

$$dv = v - v_0$$

Donde v es la velocidad en el momento y v_0 la velocidad inicial

Así nuestra ecuación de la aceleración queda de la siguiente forma:

$$a = \frac{v - v_0}{dt}$$

Debido a que el movimiento parte desde cero la velocidad inicial es cero por lo tanto la ecuación queda de la siguiente forma:

$$a = \frac{v}{dt}$$

Teniendo la aceleración del movimiento se obtiene la distancia recorrida y la coordenada para cada uno de los frames y se guardan en la lista de puntos.

5.2.4.2 *Movimiento Circular*

5.2.4.2.1 Uniforme

Se comenzó con el MCU Se tomó la misma ecuación de velocidad para realizar el movimiento, para obtener la distancia recorrida del punto inicial al punto final se utilizó la ecuación:

$$e = \theta r$$

Donde e es la distancia entre puntos, θ el ángulo entre las rectas y r el radio del círculo.

Cabe aclarar que para calcular el ángulo θ se necesita mover el eje de coordenadas al centro del círculo con la finalidad de encontrar más fácil la coordenada del movimiento y el valor de este mismo ángulo. Inicialmente se había tomado la siguiente ecuación para encontrar el ángulo entre rectas:

$$\theta = \tan^{-1} \frac{m_1 + m_2}{1 + (m_1 m_2)}$$

Donde m_1 representa la pendiente de la recta que pasa por el punto inicial y el centro mientras que m_2 es la pendiente de la recta que pasa por el punto final y el centro; sin embargo, el ángulo que se devolvía no era el deseado ya que el ángulo que devuelve esta ecuación es el que se observa en las siguiente figura:

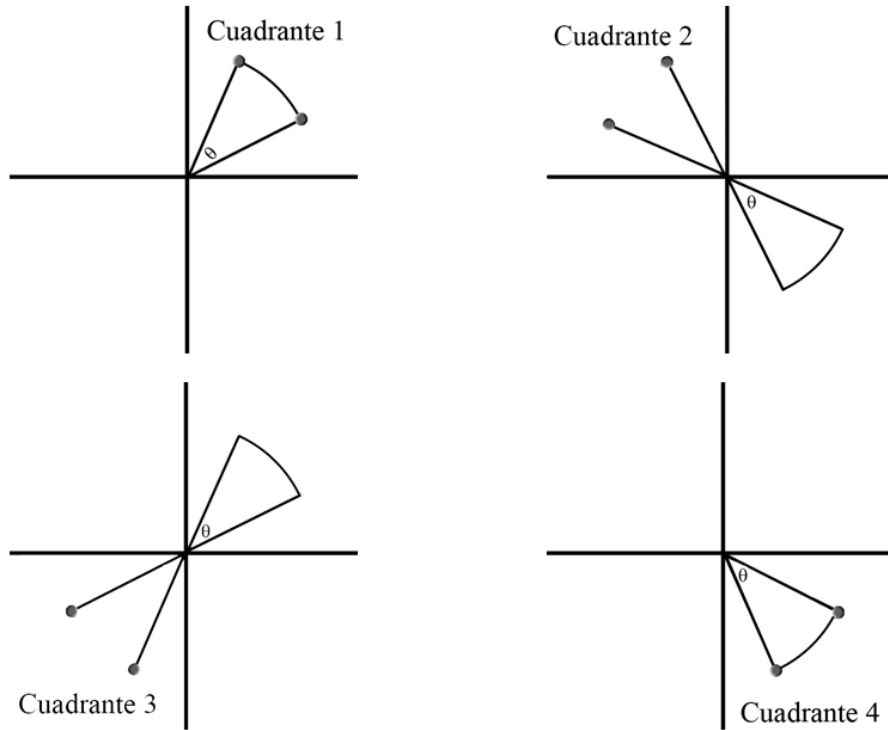


Figura 5.13 Ángulos entre rectas según el cuadrante

Como se puede observar el ángulo no es el deseado y esto afectará a la utilización de las ecuaciones posteriores por lo que se optó por calcular el ángulo que tiene cada recta con la siguiente formula:

$$\varphi = \tan^{-1} \frac{y}{x}$$

Nombrando al ángulo de la recta que pasa por el punto inicial y el centro como φ_1 y al ángulo de la recta que pasa por el punto final y el centro como φ_2 . Teniendo estos dos ángulos podemos sacar el ángulo entre rectas de la siguiente forma:

$$\theta = \varphi_2 - \varphi_1$$

La figura 5-14 explica mejor la naturaleza de esta ecuación:

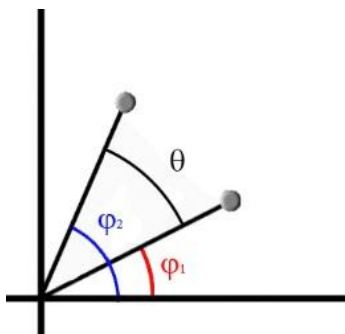


Figura 5.14 Ángulos desde el eje

Calculado el ángulo, sustituimos en la ecuación de la distancia entre puntos para obtener la distancia recorrida y así mismo hacer para la ecuación de la velocidad y obtener la misma.

Para encontrar las coordenadas de los puntos que describirán el movimiento es necesario volver a aplicar la ecuación con la que se obtuvo la distancia recorrida, sólo que esta vez necesitamos el ángulo entre la recta del punto inicial y la recta del punto que se desea encontrar sus coordenadas, despejando la ecuación queda de la siguiente forma:

$$\theta = \frac{e}{r}$$

Conociendo el radio r y la distancia e hasta el momento deseado. Obteniendo este ángulo y sabiendo que “ $\theta = \varphi_2 - \varphi_1$ ” donde como se había mencionado φ_1 es el ángulo de la recta que pasa por el punto inicial y el centro, deducimos que φ_2 es el ángulo de la recta que pasa por nuestro punto desconocido y el centro, para distinguirlo del ángulo de la recta que pasa por el punto final y el centro lo nombraremos φ_f .

Con el ángulo del punto desconocido podemos obtener sus coordenadas con las siguientes fórmulas:

$$x = r * \cos \varphi_f$$

$$y = r * \sin \varphi_f$$

Cabe destacar que al igual que en la primera fórmula que se ocupó para el ángulo entre rectas, el ángulo valor devuelto por estas ecuaciones no es el valor exacto deseado debido a los cuadrantes en los que se encuentren los puntos inicial y final, debido a esto se hizo una función para saber en qué cuadrante se encuentra cada punto y según el cuadrante de cada punto se sumarán 180, 360 o ningún grado a φ_f .

Finalmente teniendo ya la coordenada se agrega a la lista de puntos.

5.2.4.2.2 Uniforme Acelerado

Para este movimiento la aceleración con la que se trabajara será la aceleración tangencial, y se obtiene con la siguiente fórmula

$$a_t = \frac{dv}{dt}$$

Y así como en el movimiento rectilíneo uniforme la velocidad inicial será cero, que dando la ecuación de la siguiente forma

$$a_t = \frac{v}{dt}$$

De la misma forma que para el MCU se utilizan las mismas ecuaciones para las distancias recorridas, los ángulos de las respectivas rectas y las coordenadas que se irán anexando a la lista de puntos.

5.2.4.3 *Inversión*

Al realizar pruebas con esta clase se notó que en el movimiento circular podía existir la posibilidad de que el movimiento a lo largo del círculo fuera en dirección opuesta a como el programa la dibujaba por lo que se decidió poner un control para manipular la dirección al hacer un movimiento circular, así que, según esté indicado la dirección del movimiento el ángulo θ , entre la recta que pasa por el punto inicial y el centro y la recta que pasa por el punto final y el centro, calculado se dejará como se calculó o simplemente se cambiará por su ángulo complementario.

5.2.4.4 *Eje*

Es importante mencionar que el eje del movimiento no se puede colocar en una posición arbitraria sino que debe ser colocado a la misma distancia del punto inicial y del punto final, por lo que se realizó una función especial para calcular el punto donde se debe de poner el eje.

Antes de empezar a tratar el punto que el usuario desea como eje es necesario obtener la recta perpendicular a la recta trazada por el punto inicial y final, esta recta perpendicular estará orientada exactamente entre los puntos inicial y final y marcará donde deben de ir cualquier punto de eje. Conociendo esta recta se recibirá el punto donde el usuario quiere el eje se calculará una recta paralela a la recta de los puntos inicial y final, que pase por el punto donde el usuario dio clic, y donde se cruce con la recta perpendicular calculada anteriormente ahí se colocará el punto de eje.

5.2.5 *Estructura del proyecto*

Para tener en orden los fotogramas capturados, el proyecto está organizado en escenas. Cada escena se almacena en una carpeta nombrada igual que la escena.

Los fotogramas capturados son nombrados con el siguiente formato: La palabra “Captura” seguida del número de fotograma y la extensión “.jpg”.

Todo el proyecto es almacenado en un archivo XML, con una estructura similar a la que se describe a continuación en la Tabla 5.1.

Tabla 5.1 Estructura del archivo XML

<pre> <proyecto> <nombre>Nombre de Proyecto</nombre> <FPS>24</FPS> <escenas> <contenido> <escena> <nombre>Escena 1</nombre> <tomas> <contenido> <toma>Captura1.jpg</toma> <toma>Captura2.jpg</toma> <toma>Captura3.jpg</toma> </contenido> </tomas> </escena> </contenido> <escena> <nombre>Escena 2</nombre> <tomas> <contenido> <toma>Captura1.jpg</toma> </contenido> </tomas> </escena> </escenas> </proyecto> </pre>	<p>Nodo raíz</p> <p>Nombre del proyecto</p> <p>Fotogramas por segundo</p> <p>Nodo principal de escenas</p> <p>Nodo de una escena</p> <p>Nombre de la escena 1</p> <p>Nodo principal de las tomas</p> <p>Nombres de los archivos de las tomas dentro de la escena</p> <p>Nodo de una escena</p> <p>Nombre de la escena 2</p> <p>Nodo principal de las tomas</p> <p>Nombres de los archivos de las tomas dentro de la escena</p>
---	--

Fuente: Elaboración propia

Tenemos una clase para el manejo de los proyectos que se encarga de abrir el archivo XML, interpretar y cargar la información contenida en dicho archivo de forma estructurada para el fácil acceso en otros módulos del sistema y generar un objeto XML con la información contenida en la misma clase para posteriormente ser almacenado.

Esta clase contiene, además, métodos para añadir y eliminar escenas, para añadir uno o varios fotogramas a una determinada escena y para eliminar los fotogramas de una escena determinada.

Para distinguir entre los archivos XML que no son proyectos de Stop Motion, se utilizó la extensión PSM (Proyecto Stop Motion) como extensión propia del sistema.

5.2.6 Visualización de la cámara

En este apartado se describe todo lo relacionado con el área de visualización.

5.2.6.1 Visualización de imágenes

OpenCV nos brinda la posibilidad de obtener imágenes de la cámara utilizando un objeto de la clase Capture [13]. La manera de mostrar el objetivo de la cámara en tiempo real es creando un método que solicite la imagen al objeto Capture con el método QueryFrame y en caso de ser necesario aplicar los efectos que el usuario haya seleccionado y actualizar la imagen en el componente de la interfaz donde se va a visualizar. Posteriormente añadimos este método al evento de inactividad de la aplicación de tal manera que siempre que la aplicación no este procesando información será solicitado y mostrada una nueva imagen de la cámara.

5.2.6.2 Mostrar área Segura

Gracias a las estructuras Rectangle y LineSegment2D (Propio de openCV) podemos definir un rectángulo que representa el marco del Área Segura, y la cruz que representa el centro de la imagen, y dibujarlos sobre la imagen utilizando el método Draw de la clase Image, que nos brinda OpenCV [13].

5.2.6.3 Papel Cebolla

Se crea el efecto de papel cebolla realizando una suma ponderada [14] entre la imagen capturada por la cámara y uno o hasta 3 frames capturados previamente. Los diferentes pesos que se dan a la suma pondera cambian dependiendo la cantidad de fotogramas a mostrar con el efecto de papel cebolla. Cuando solamente se muestra un fotograma, las ponderaciones son del 50-50%. Cuando se muestran dos fotogramas, primero se hace una suma ponderada entre los dos fotogramas, con una ponderación de 60-40%, utilizando este resultado se vuelve a hacer otra suma ponderada con la imagen capturada y pesos de 50-50%. En el último caso, cuando se muestran tres fotogramas, se repite el proceso, sumando el antepenúltimo y penúltimo fotograma capturado, con pesos de 40-60%, después sumando este resultado y el último fotograma capturado con pesos 50-50%. Finalmente este resultado se suma a la imagen capturada con pesos 50-50%. Todas estas ponderaciones se calcularon en base a pruebas que se fueron haciendo conforme al desarrollo de esta parte del sistema.

5.2.6.4 Dibujado de los puntos

Para el dibujado de los puntos clave (inicial, final y eje), se utiliza el evento del clic sobre el componente donde se visualizan las imágenes de la cámara.

Inicialmente, se obtenían las coordenadas del punto donde el usuario hacía clic y se dibujaban los puntos directamente en el componente. Sin embargo, al hacer pruebas, se descubrió que al cambiar el tamaño de la ventana, los puntos se movían de su lugar, quedando incluso afuera de la imagen de la cámara.

Para solucionar esto, se pensó en dibujar los puntos no sobre el componente, sino sobre la imagen. El siguiente problema que se presentó fue que el punto donde el usuario hacía clic

no coincidía con el punto dibujado sobre la imagen, debido a la diferencia de tamaños entre el componente que recibe el clic y la imagen donde se dibuja.

Por ello, se realizó el cálculo del punto “real” en base al tamaño del componente C , al tamaño de la imagen de la cámara a tamaño real I y en escala P .

Se hace una división entre las dimensiones de la imagen (representadas como I_{ancho} e I_{alto}) para obtener la relación de aspecto de la imagen R como se muestra en la siguiente ecuación:

$$R = \begin{cases} \frac{I_{alto}}{I_{ancho}} & C_{alto} > C_{ancho} \\ \frac{I_{ancho}}{I_{alto}} & C_{alto} < C_{ancho} \end{cases}$$

Una vez que se tiene esta relación, se toma el lado más pequeño del componente de visualización (C_{ancho} o C_{alto}), el cuál representa ya sea la dimensión en x o en y de la imagen a escala, esto se explica a continuación:

$$\text{Si } C_{alto} > C_{ancho} \text{ entonces: } P_{ancho} = C_{ancho}$$

$$\text{si no: } P_{alto} = C_{alto}$$

Este valor es multiplicado por la relación R obtenida anteriormente y así se obtiene la otra dimensión de la imagen a escala ver la siguiente ecuación

$$\text{Si } C_{alto} > C_{ancho} \text{ entonces: } P_{alto} = R \times P_{ancho}$$

$$\text{si no: } P_{ancho} = R \times P_{alto}$$

Una vez que se tiene el tamaño de la imagen a escala (expresado por $P_{ancho} \times P_{alto}$), se realiza una división entre el ancho de la imagen a escala P_{ancho} y su equivalente en la imagen real I_{ancho} para obtener un valor P que nos indica la proporción de tamaño entre la imagen real y la imagen escalada, este valor se obtiene con la siguiente fórmula:

$$P = I_{ancho} / P_{ancho}$$

Por medio de una resta, se calcula el espacio que sobra en el componente de visualización tanto vertical como horizontalmente para ver qué tan desfasados estarán los puntos en la imagen al momento de dibujarlos. El cálculo del desfase se muestra en las siguientes ecuaciones:

$$d_x = C_{ancho} - P_{ancho}$$

$$d_y = C_{alto} - P_{alto}$$

Finalmente, para obtener el punto real (R_x, R_y) a cada componente del punto donde se dio clic (M_x, M_y) se le resta el desfase y el resultado es multiplicado por el valor de la proporción obtenida anteriormente como se muestra en las siguientes ecuaciones:

$$R_x = P(M_x - d_x)$$

$$R_y = P(M_y - d_y)$$

Los puntos son almacenados en la clase física, la cual, al llamar a sus métodos para calcular los puntos intermedios, nos regresa una lista de puntos que se pintarán sobre la imagen con el método Draw de forma similar a como se hacía con el área segura.

Al pintar directamente sobre la imagen, se garantiza que los puntos se queden en su lugar sin importar de qué modo se altere el tamaño de la ventana.

5.2.7 Captura de fotogramas

Se cuenta con un botón de captura que realizará la captura de 1 fotograma cada que el usuario lo accione y también se tiene un botón de captura rápida que al ser accionado por el usuario comenzará a capturar fotogramas a la velocidad indicada en el proyecto hasta que el usuario lo accione por segunda vez.

5.2.7.1 Captura individual

El botón de captura le pide al objeto Capture un fotograma mediante el método QueryFrame [13], el cual, después de ser aplicado el efecto de iluminación que desee el usuario, será almacenado en la carpeta de la escena que se encuentre seleccionada.

5.2.7.2 Captura rápida

Se calcula el tiempo de espera entre cada fotograma y, con el evento Tick de un objeto Timer, se realiza una captura en cada intervalo de tiempo de forma indefinida hasta que el usuario lo indique. La manera de realizar las capturas es idéntica a como se hacen con el botón de captura individual.

5.2.8 Control de iluminación

Se añadió un control que permite variar la iluminación de la imagen que captura la cámara. Esto se logra fácilmente mediante el método Mul de la clase Image que nos proporciona OpenCV [13]. Dichos controles de iluminación incrementan o decrementan una variable que indica el factor de iluminación que se aplicará al método Mul que permite aclarar u oscurecer la imagen.

5.2.9 Previsualización de la animación

Se creó un Form llamado PreviewWindow, el cual contiene un área de previsualización y controles para reproducir, detener, avanzar y retroceder fotogramas, repetición y selección de la velocidad.

Al crearse este objeto, se carga una lista de los fotogramas que componen la escena seleccionada en la ventana principal. Nuevamente con la ayuda de un objeto Timer, se va cambiando la imagen que se muestra en el área de previsualización cada cierto tiempo de acuerdo a la velocidad de fotogramas por segundo seleccionada por el usuario.

Se estableció una bandera de valor booleano para determinar si la animación debe repetirse indefinidamente y los controles de reproducción, pausa, avanzar y retroceder modifican el índice que indica cuál es la imagen que se mostrará en el área de pre visualización.

Para el cambio de velocidad, se tiene un combo box con las velocidades en fotogramas por segundo más comunes, del cual se recupera el valor y se hace un sencillo cálculo para determinar el intervalo de tiempo en el que se deben ir cambiando las imágenes.

6 Pruebas

La metodología de desarrollo XP requirió de la realización de pruebas continuas al sistema, mismas que se fueron haciendo paralelamente conforme el desarrollo del proyecto para garantizar que no hubiera errores por parte del usuario; adicionalmente, se realizaron pruebas con el equipo de animación Lobo's Films, donde el Lic. Eduardo Campos y su equipo nos ayudaron a realizar un par de cortos con nuestro sistema. A continuación se describen cada una de las pruebas realizadas.

6.1 Pruebas al sistema

Para probar el sistema desarrollado, se hicieron 4 pequeñas animaciones de prueba, cada una de estas se realizó con nuestro sistema y con el programa *DragonFrame* antes conocido como *Dragon Stop Motion*.

La primera se hizo con el propósito de probar el Movimiento Circular Uniformemente Acelerado, para ello se utilizó una figura articulada de 8 pulgadas. Se hizo la animación del movimiento de su brazo izquierdo, levantándolo de la cintura hasta la altura de su cabeza. La prueba realizada con *DragonFrame* no utilizó ninguna referencia para colocar el brazo del muñeco. La Figura 6.1 muestra una captura de la animación de prueba.



Figura 6.1 MCUA utilizando *DragonFrame*

Se realizó la misma animación con nuestro sistema pero utilizando los puntos calculados por la clase física. La Figura 6.2 muestra nuestro sistema con los puntos de referencia.

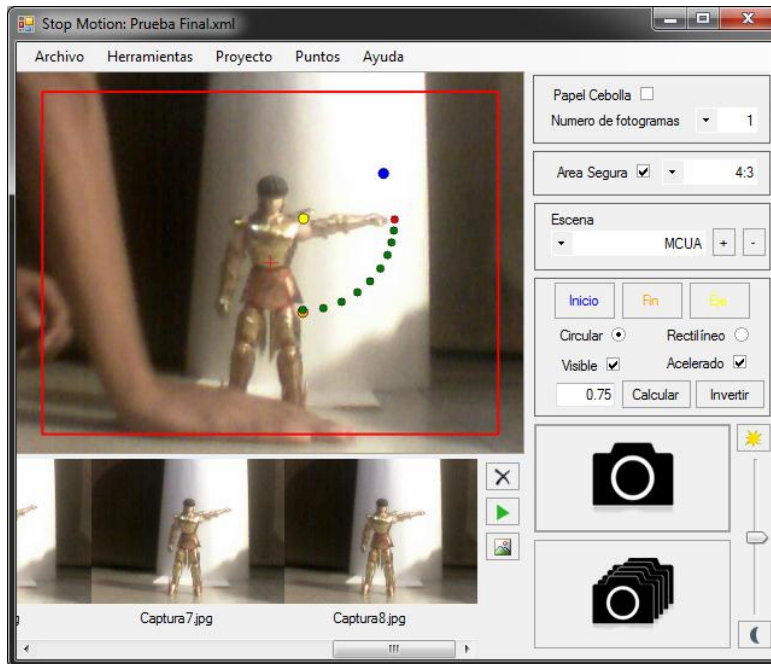


Figura 6.2 MCUA con nuestro sistema

Los resultados en cuanto a tiempo se muestran en la Tabla 6.1

La segunda fue para probar el Movimiento Circular Uniforme, para ello se utilizó la misma figura articulada. Se hizo la animación del movimiento de su brazo izquierdo formando una trayectoria circular, estando el muñeco de perfil. Nuevamente, la prueba realizada con *DragonFrame* no utilizó ninguna referencia para colocar el brazo del muñeco. La Figura 6.3 Muestra una captura de la animación de prueba utilizando dicho software.



Figura 6.3 MCU con DragonFrame

En la Figura 6.4 se muestra la misma toma pero en nuestro sistema utilizando los puntos guía.

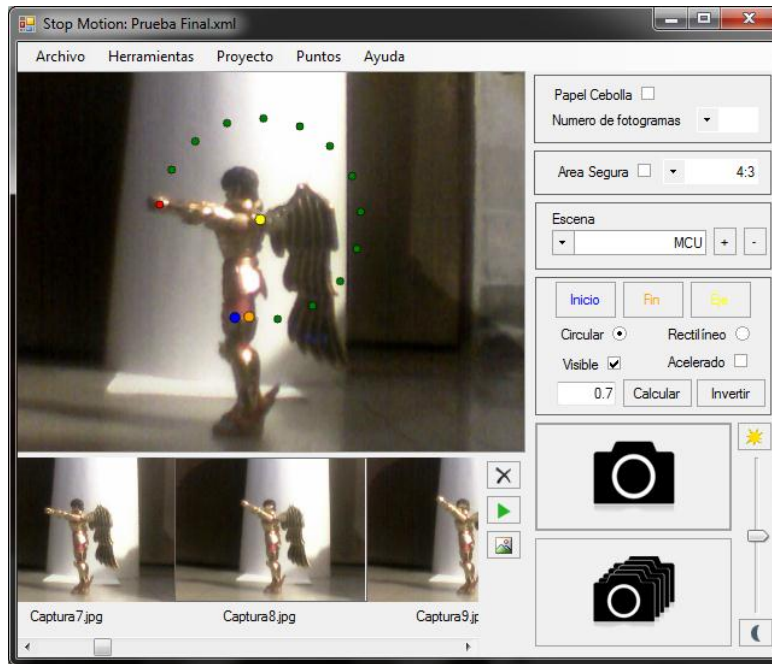


Figura 6.4 MCU con nuestro sistema

La tercera prueba es sobre el movimiento Rectilíneo Uniformemente Acelerado, para ello se hizo la animación de una naranja pasando de un lado a otro de la cámara de forma acelerada. La Figura 6.5 muestra la toma realizada con *DragonFrame* sin ninguna referencia.

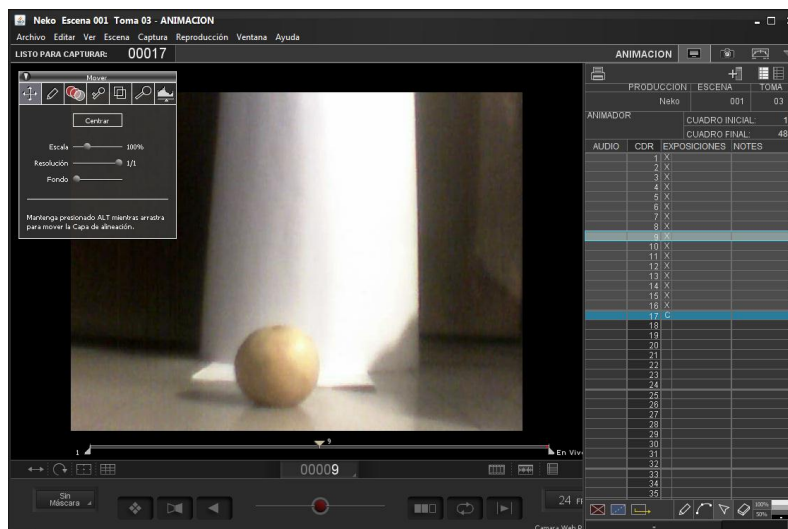


Figura 6.5 MRUA con Dragon Frame

En la Figura 6.6 se muestra la misma toma pero con nuestro sistema y con ayuda de los puntos que el mismo sistema nos proporciona, para poder utilizar los puntos de forma efectiva, se pintó un rostro en la naranja para usarlo de referencia.

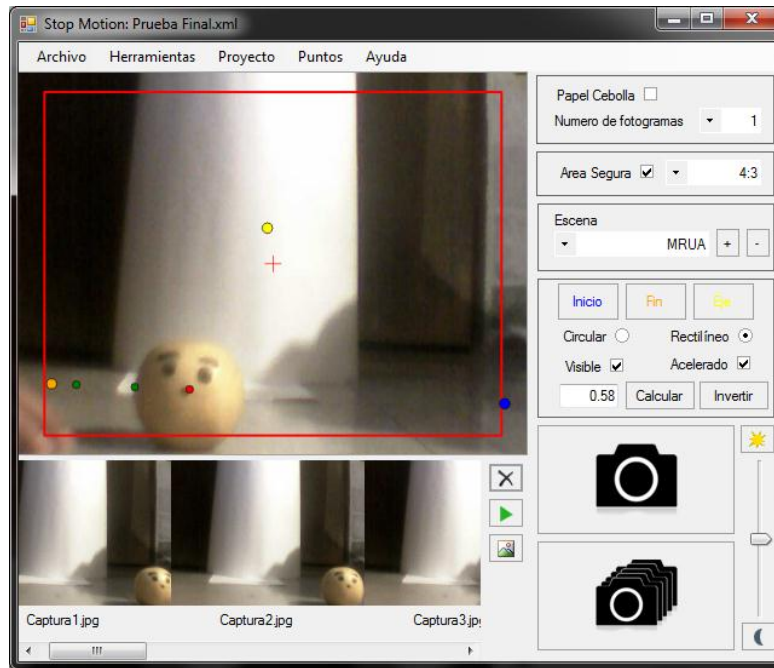


Figura 6.6 MRUA con nuestro sistema

La cuarta prueba es muy similar a la anterior, sólo que en esta ocasión, la naranja se mueve a velocidad constante. La Figura 6.7 muestra cómo se realizó la toma con *DragonFrame*.

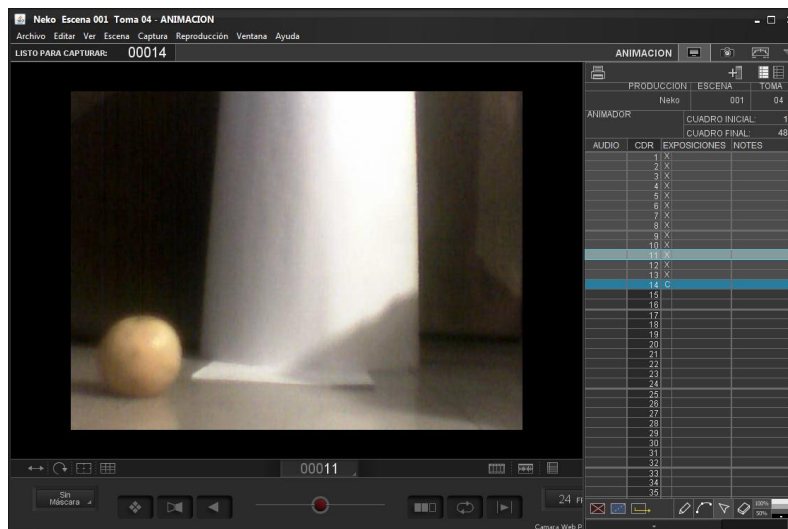


Figura 6.7 MRU con Dragon Frame

En nuestro sistema, se calcularon los puntos del movimiento y con ayuda de ellos y tomando como referencia la boca pintada en la naranja, se pudo realizar con mayor precisión la animación.

La Figura 6.8 muestra como se realizó la animación con nuestro sistema.

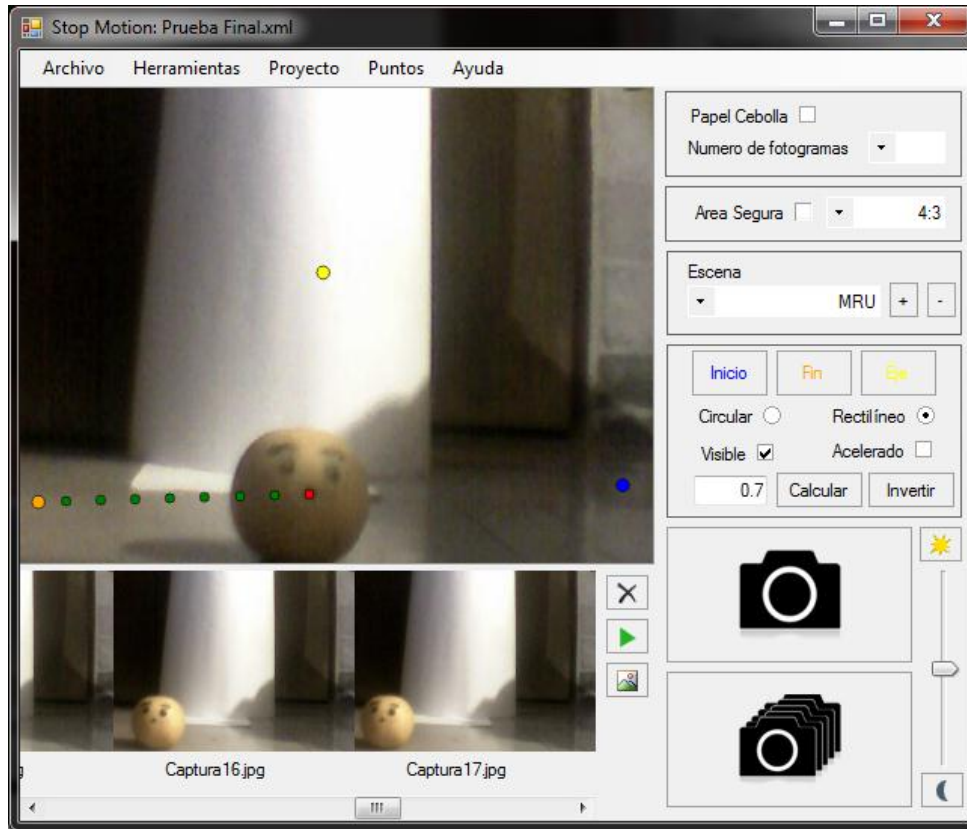


Figura 6.8 MRU con nuestro sistema

En la Tabla 6.1 se muestran los resultados comparando los tiempos de realización de las pruebas con nuestro sistema y con *DragonFrame*.

Tabla 6.1 Resultados

	Nuestro sistema (minutos)	Dragon Frame (minutos)	Fotogramas	Duración de la animación (segundos)
Movimiento circular acelerado	6:00	7:00	18	0.75
Movimiento circular uniforme	6:40	15:00	24	1.00
Movimiento rectilíneo uniforme	3:00	2:40	17	0.70
Movimiento rectilíneo acelerado	2:00	3:00	14	0.58

Fuente: Elaboración propia

La diferencia en los tiempos se debe a errores que cometíamos al momento de realizar las pruebas; sin embargo, al utilizar nuestro sistema es mucho más fácil ubicar los puntos de movimiento y por esta razón la animación se realiza con una mayor calidad y se producen menos errores. Se concluye que la aplicación *DragonFrame* no indica el centro de la pantalla y es muy complejo su uso. También observamos que para hacer una toma de un segundo se requiere de bastante tiempo, aunque esto depende en mayor parte de las habilidades y experiencias del animador. En general se obtuvieron menores tiempos de animación y mejores resultados con el uso de nuestro sistema como se puede observar en la Figura 6.9 por lo que nuestro sistema presenta un mayor rendimiento y mejor calidad.

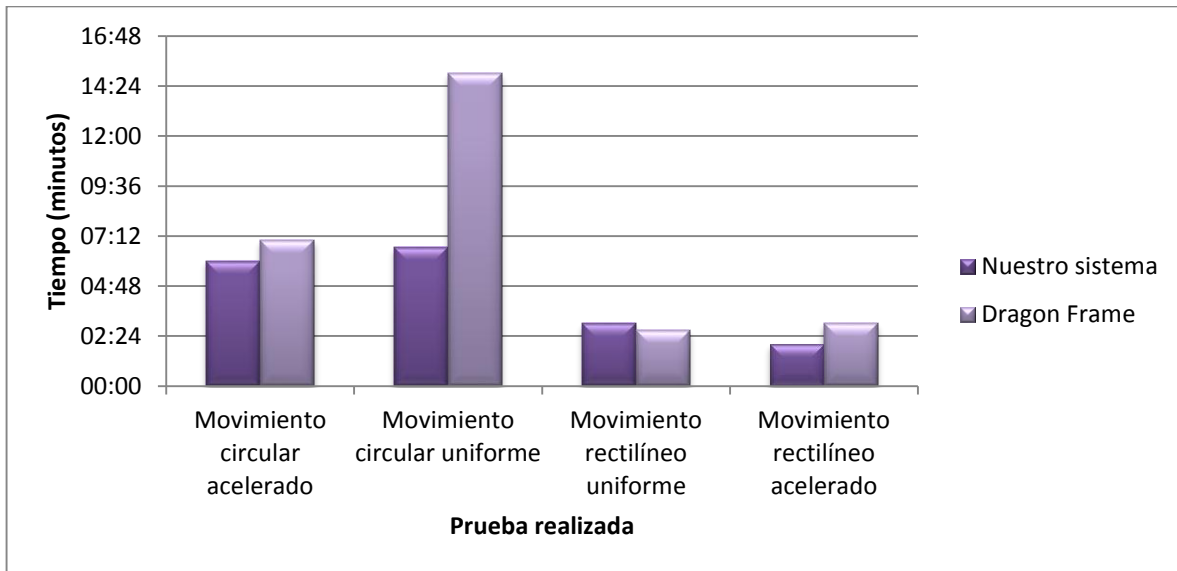


Figura 6.9 Comparativa de tiempos para realizar las pruebas

6.2 Pruebas con los usuarios

En el estudio de animación de Lobo's Films, realizamos dos breves animaciones con nuestro sistema: animación de un brazo en movimiento y animación de un muñeco que se coloca la cabeza.

Para la primera prueba, utilizamos los puntos como una referencia para colocar el brazo de la armadura articulada. En la Figura 6.10 se muestra una captura de pantalla del sistema durante la realización de esta animación.

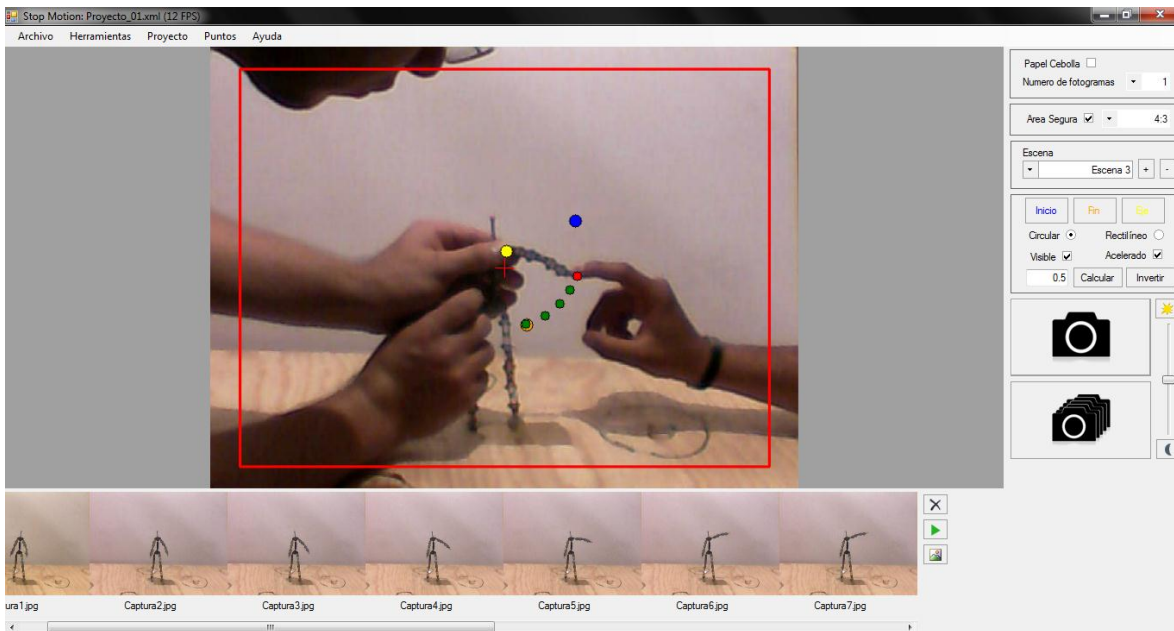


Figura 6.10 Prueba del sistema en un estudio de animación

Para la segunda prueba, una de las herramientas utilizadas que resultó de mucha utilidad fue el efecto de papel cebolla, el cual nos permitió volver a colocar la figura en la posición que se encontraba después de que accidentalmente se cayó y se pudo continuar con la captura de los fotogramas.

En esta prueba, se utilizaron los puntos para mover una patineta en línea recta con velocidad constante que corre detrás del muñeco. En la Figura 6.11 se puede apreciar cómo es que se utilizó esta herramienta para la creación de dicha animación.

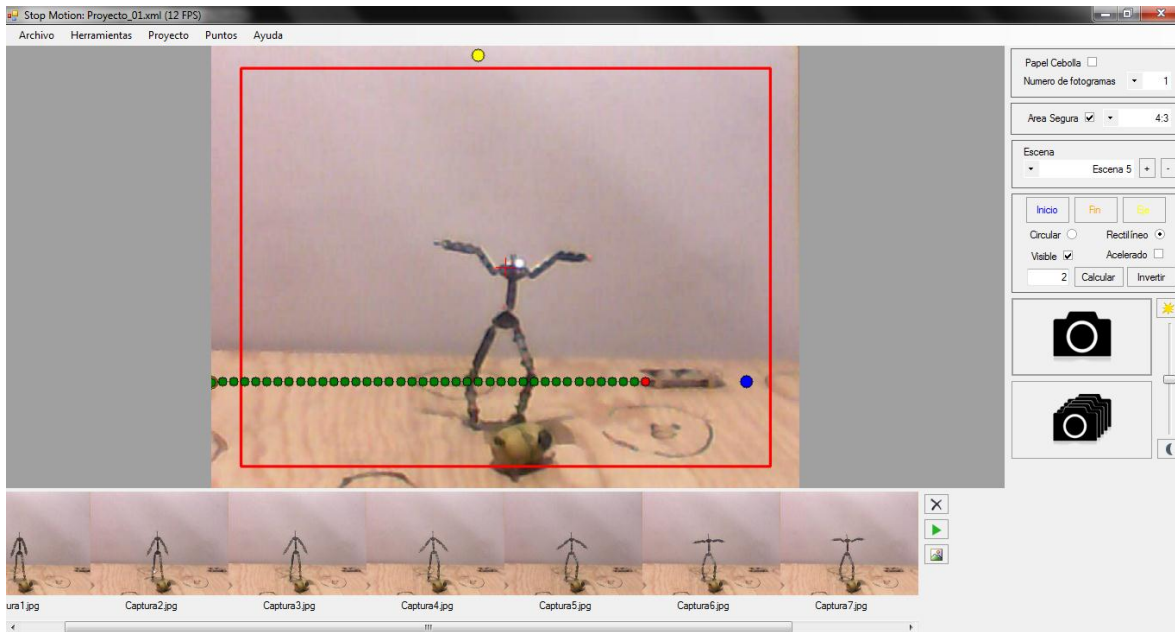


Figura 6.11 Uso de puntos para animar una patineta con velocidad constante

Ambas pruebas fueron realizadas a una velocidad de 12 fotogramas por segundo, capturando alrededor de 60 cuadros por animación, lo que equivale a 5 segundos aproximadamente. El realizar estas pruebas nos tomó un tiempo aproximado de 50 minutos desde que se montó el pequeño set hasta que se capturó el último fotograma.

Durante el desarrollo de estas pruebas, los usuarios nos hicieron la observación de que el tiempo para abrir el programa, montar la cámara y empezar la captura de imágenes con nuestro sistema es mucho más corto que el tiempo que les tomaba a ellos hacer la toma de forma manual utilizando únicamente la cámara. En varias ocasiones nos pidieron mostrarles el efecto de papel cebolla para usarlo como referencia y colocar la figura en su lugar.

Como comentario final, los animadores nos dijeron que el sistema es bastante funcional para usuarios aficionados, que cumple con todo lo necesario para realizar la captura de fotogramas, que se reduce el tiempo para empezar a capturar las imágenes y que las herramientas que tiene resultan de mucha utilidad. Sin embargo, nos hicieron también la observación de que es necesario utilizar más puntos de referencia para hacer movimientos más complejos como hacer que la armadura simule caminar..

7 Conclusiones

En el desarrollo de este proyecto pudimos aplicar, finalmente, en un caso real muchos de los conocimientos de física y geometría adquiridos a lo largo de la carrera.

Pudimos experimentar la rapidez con la que se desarrolla un proyecto utilizando la metodología XP que combinado con el uso de herramientas en la nube, como el almacenamiento de archivos, documentos en línea, y medios de comunicación, nos facilita el trabajo aun estando separados y disminuye considerablemente el tiempo de desarrollo en comparación con otras metodologías.

Logramos comprobar que las habilidades desarrolladas durante toda nuestra estancia en la escuela, nos ayudan profesionalmente, permitiéndonos dominar nuevas tecnologías en base a todo el conocimiento acumulado, como lo es la tecnología XML, que nos permitió ordenar de manera eficaz información y recuperarla posteriormente con la misma sencillez.

Se alcanzó el objetivo de apoyar en la realización de películas Stop Motion, disminuyendo los tiempos de captura de las escenas y mejorando su fluidez, al decirle al animador donde poner el objeto en cada fotograma.

En cuanto a la calidad de la animación, en las pruebas se pudo observar que el movimiento se realizaba de manera más fluida si se siguen los puntos de referencia que calcula nuestro software, además de que era más fácil ubicar la figura siguiendo estos puntos. En base a esto, podemos concluir que nuestro sistema puede ayudar mucho a los animadores aficionados y a las personas que no tienen experiencia haciendo Stop Motion.

8 Trabajo a futuro

El presente trabajo puede extenderse para incluir otras funcionalidades como la posibilidad de seleccionar múltiples objetos u objetivos en una sola escena y que cada uno pueda representar diferentes tipos de movimiento o tener varios ejes, si se requiere, además de coordinarlos o sincronizarlos para crear movimientos más complejos como lo es el movimiento de una persona al caminar. También se ha propuesto marcar la rotación de un objeto sobre su propio eje de forma análoga a como se realizó el cálculo de los puntos. Otra funcionalidad que podría incluirse es la posibilidad de generar el archivo de video una vez que se tengan todos los fotogramas.

Finalmente, con el propósito de ampliar el mercado de nuestro sistema, se podría incluir el soporte para diferentes tipos de cámaras, incluyendo las profesionales.

También podría diseñarse un control de hardware para tomar los fotogramas de forma remota, controlar la iluminación y apagar la pantalla o un controlador para manipular la iluminación del set.

9 Referencias

- [1] B. J. C. Purves, Stop Motion Passion, Process and Performance, ELSEVIER, 2008.
- [2] T. Gasek, Frame by Frame: The guide to non-traditional animation techniques, Focal Press, 2011.
- [3] Pharos Productions, "The Art of Stop Motion Animation," 2005. [Online]. Available: <http://pharosproductions.com>. [Accessed 23 Abril 2012].
- [4] K. A. Priebe and H. Selick, The advanced art of Stop-Motion animation, Course Technology, 2011.
- [5] I. D. Aronson, DV Filmmaking, O'Reilly Media, Inc., 2006.
- [6] The Digital Bits, «<http://www.thedigitalbits.com>,» The Digital Bits, 6 Marzo 2000. [En línea]. Available: <http://www.thedigitalbits.com/articles/anamorphic/aspectratios/widescreenorama.html>. [Último acceso: 23 Abril 2012].
- [7] Wavelength Media, «<http://www.mediacollege.com>,» Wavelength Media, 1995. [En línea]. Available: <http://www.mediacollege.com/video/aspect-ratio/>. [Último acceso: 23 Abril 2012].
- [8] D. F. Cheshire, The book of movie photography: the complete guide to better moviemaking, Knopf, 1979.
- [9] Resnick, Halliday y Krane, Fisica 1, Editorial Continental, 2001.
- [10] B. Venners and B. Eckel, "Generics in C#, Java, and C++," 26 Enero 2004. [Online]. Available: <http://www.artima.com/intv/genericsP.html>. [Accessed 27 Agosto 2011].
- [11] S. Priolo, «Programacion Extrema,» [En línea]. Available: ftp://190.5.199.3/jjurado/Mejora%20de%20Procesos/XP_6JAI.pdf. [Último acceso: 14 Octubre 2011].
- [12] S. Ian, Ingenieria de Software, Pearson Addison Wesley, 2005.
- [13] OpenCV, «OpenCV Documentation,» 02 abril 2011. [En línea]. Available: <http://docs.opencv.org/>. [Último acceso: 22 febrero 2012].
- [14] G. Pajares Martin-Sanz, Vision por computador imágenes digitales y aplicaciones, Apha-Omega, 2008.

10 Glosario

Filmografía: Conjunto de películas cinematográficas que tienen una característica común, como pertenecer a un mismo actor o director, tratar un mismo tema, ser de una época determinada, etc.

Cortometraje: Película de cine cuya duración no es mayor de 35 minutos

Cineasta: persona que se encarga de la dirección y realización de una película o del arte conocido como cine. El cineasta es quien tiene a cargo a todos los actores y trabajadores del rubro ya que es él quien da las órdenes, indica posiciones, formas de representar determinadas situaciones, dirigir la edición y los arreglos finales.

Resolución: Nitidez con que se percibe una imagen observada por un instrumento óptico o la imagen formada en una película fotográfica o en la televisión

Captura: imagen tomada a través de una computadora con el objetivo de registrar los elementos visibles, tanto del monitor como de cualquier otro de los dispositivos de salida visual.

Iluminación: técnica se refiere al conjunto de dispositivos que se instalan para producir ciertos efectos luminosos, tanto prácticos como decorativos, factor expresivo y funcional que ayuda a crear continuidad entre los distintos planos y a aumentar la expresión artística de la imagen.

Traslúcido: Se aplica al cuerpo que deja pasar la luz pero que no permite ver con nitidez a través de él: cristal translúcido.

Proyectil: objeto lanzado en el espacio por la acción de una fuerza.

Inerte: aquello que se caracteriza por la falta de vida, de movilidad o bien por su inutilidad

Climatológica: Condiciones propias de cada clima.

Subsanar: Resarcir o remediar un defecto, un daño o un error, una dificultad o un problema.