

**UNIVERSIDAD AUTONOMA DE MADRID**

**ESCUELA POLITECNICA SUPERIOR**



## **PROYECTO FIN DE CARRERA**

**HERRAMIENTAS DE APOYO A LA FUSIÓN DE  
SECUENCIAS DE VÍDEO DE ALTA VELOCIDAD PARA  
AUMENTAR LA RESOLUCIÓN TEMPORAL O EL  
CAMPO DE VISIÓN EFECTIVO**

**Eduardo Jiménez Sánchez**

**JULIO 2008**

**HERRAMIENTAS DE APOYO A LA FUSIÓN DE  
SECUENCIAS DE VÍDEO DE ALTA VELOCIDAD PARA  
AUMENTAR LA RESOLUCIÓN TEMPORAL O EL  
CAMPO DE VISIÓN EFECTIVO**

**AUTOR: Eduardo Jiménez Sánchez  
TUTOR: Jesús Bescós Cano**

**Grupo de Tratamiento de Imágenes  
Dpto. de Ingeniería Informática  
Escuela Politécnica Superior  
Universidad Autónoma de Madrid  
Julio 2008**

## **PROYECTO FIN DE CARRERA**

**Título:** *Herramienta de apoyo a la fusión de secuencias de vídeo de alta velocidad para aumentar la resolución temporal o el campo de visión efectivo.*

**Autor:** Eduardo Jiménez Sánchez

**Tutor:** Jesús Bescós Cano

**Tribunal:**

**Presidente:** José María Martínez Sánchez

**Vocal:** Miguel Ángel García García

**Vocal secretario:** Jesús Bescós Cano

**Fecha de lectura:**

**Calificación:**

## Palabras clave

---

Multicámaras, aumento de resolución, aumento de la tasa de cuadro, corrección radial, corrección de perspectiva, homografías, ajuste de contraste, Matlab .

## Resumen

---

Las necesidades actuales de la ciencia hacen que sea necesario capturar eventos que ocurren a muy altas velocidades para su posterior estudio. En estos estudios se emplean cámaras de alta velocidad.

En este proyecto se analiza la posibilidad de emplear sistemas multicámara, utilizando cámaras de vídeo de alta velocidad, para incrementar la resolución temporal o espacial de una secuencia de vídeo, solución notablemente más económica que la de utilizar una cámara más sofisticada, a veces inviable. El proyecto presenta la estructura de un sistema completo para el que se han desarrollado aplicaciones de configuración y manejo.

Adicionalmente se han implementado herramientas para corregir las diversas distorsiones presentes en las secuencias que genera el sistema, distorsiones inherentes al uso de varias cámaras en vez de una sola. Finalmente, se ha desarrollado una aplicación de análisis cinemático para estimar las magnitudes físicas (velocidad , distancias ... ) de los objetos en movimiento que aparecen en la secuencia bajo estudio.

## Abstract

---

Actual science research requires to capture events that happened at very high speeds for its posterior study. In these cases, high-speed cameras are employed.

The present project analyses the possibility of using multicamera systems constituted by High-speed cameras, in order to increase a sequence's spatial or temporal resolution. The proposed solution is cheaper than using a more specialized camera, which in many cases becomes not feasible. The project shows the structure of a complete system and some applications that have been developed for its configuration and operation.

Additionally, some tools have been implemented to correct different distortions in the image sequences that the system introduces inherently by the use of combinations of several cameras. Finally, an application for cinematic analysis has been developed to estimate some physical magnitudes (i.e. speed , distance ...) of the motion objects that appear in the sequence under study.

## ***Agradecimientos***

***A la escuela y especialmente a los miembros del GTI y a Jesús quienes me han ayudado en todo lo posible para llevar a cabo este trabajo***

***A mi familia por haberme dado la oportunidad de haber llegado hasta aquí.***

***A mi niña por haberme aguantado mis altibajos.***

***A ARIES S.A. , especialmente a Ricardo e Iván sin los cuales no hubiera sido posible llevar este proyecto a la práctica.***

## INDICE DE CONTENIDOS

<b>Palabras clave</b> .....	<b>4</b>
<b>Resumen</b> .....	<b>4</b>
<b>Abstract</b> .....	<b>4</b>
<b>1. Introducción</b> .....	<b>2</b>
<b>1.1. Motivación</b> .....	<b>2</b>
<b>1.2. Objetivo</b> .....	<b>2</b>
<b>1.3. Estructura de la memoria</b> .....	<b>3</b>
<b>2. Antecedentes y planteamiento</b> .....	<b>4</b>
<b>2.1. Planteamiento del objetivo</b> .....	<b>2</b>
2.1.1. Fundamentos del aumento de la resolución temporal.....	2
2.1.2. Problemática asociada al proceso de aumento de la resolución temporal.....	4
Sincronización .....	5
2.1.3. Fundamentos del aumento de la resolución espacial .....	5
2.1.4. Problemática asociada al proceso de aumento de la resolución espacial.....	6
Sincronización .....	7
<b>2.2. Arquitectura del sistema</b> .....	<b>7</b>
2.2.1. Cámaras .....	7
2.2.2. Sistema <i>AumentoTasa</i> : elementos y montaje.....	8
2.2.3. Sistema <i>AumentoCampo</i> : elementos y montaje .....	10
<b>2.3. Diseño software</b> .....	<b>11</b>
<b>2.4. Conclusiones</b> .....	<b>11</b>
<b>3. Desarrollo software</b> .....	<b>13</b>
<b>3.1. Especificaciones iniciales</b> .....	<b>13</b>
3.1.1. Descripción y objetivos iniciales .....	13
3.1.2. Requisitos .....	14
3.1.2.1. Asistente de alineamiento y ajuste .....	14
3.1.2.2. Herramienta de captura y montaje .....	14
3.1.2.3. Herramienta de análisis .....	15
<b>3.2. Herramienta de desarrollo: Matlab 7.0.1</b> .....	<b>15</b>
3.2.1. Entorno de desarrollo.....	15
3.2.1.1. GUIDE (Graphical User Interfaz Design Environment) .....	16
3.2.1.2. SDK de las cámaras .....	17
3.2.1.3. Image Processing Toolbox.....	19
<b>3.3. Descripción de la aplicación</b> .....	<b>19</b>
3.3.1. Introducción.....	19
3.3.2. Modo <i>AumentoTasa</i> : Asistente de alineamiento y ajuste .....	20
3.3.3. Modo <i>AumentoTasa</i> : Herramienta de captura y montaje.....	21
3.3.4. Modo <i>Panorámic</i> : Asistente de alineamiento y ajuste.....	22
3.3.5. Modo <i>AumentoCampo</i> : Herramienta de captura y montaje .....	23
3.3.6. <i>Cinematic Analysis</i> .....	24
<b>3.4. Funcionamiento de la aplicación</b> .....	<b>25</b>
3.4.1. Introducción.....	25
3.4.2. Flujo de las aplicaciones.....	26
3.4.3. Variables y estructuras utilizadas .....	28
3.4.3.1. Modo <i>AumentoTasa</i> y <i>AumentoCampo</i> .....	28
3.4.3.2. Aplicación de <i>Cinematic Analysis</i> .....	30

3.4.4.	AumentoTasa.....	30
3.4.4.1.	Descripción del sistema de archivos .....	30
3.4.4.2.	Diagrama del asistente de alineamiento y ajuste .....	31
3.4.4.3.	Diagrama de la herramienta de captura y montaje .....	33
3.4.5.	<i>Panorámic</i> .....	36
3.4.5.1.	Descripción del sistema de archivos .....	36
3.4.5.2.	Diagrama del asistente de alineamiento y ajuste .....	37
3.4.5.3.	Diagrama de la herramienta de captura y montaje .....	38
3.4.6.	<i>Cinematic Analysis</i> .....	40
3.4.6.1.	Descripción del sistema de archivos .....	40
3.4.6.2.	Diagrama funcional.....	41
<b>3.5.</b>	<b>Problemas y trabajo futuro.....</b>	<b>45</b>
<b>3.6.</b>	<b>Conclusiones.....</b>	<b>45</b>
<b>4.</b>	<b><i>Desarrollo de algoritmos de corrección de los errores asociados al proceso de captación y fusión.....</i></b>	<b>47</b>
<b>4.1.</b>	<b>Requisitos iniciales.....</b>	<b>47</b>
<b>4.2.</b>	<b>Modo AumentoTasa .....</b>	<b>48</b>
<b>4.3.</b>	<b>Modo AumentoCampo .....</b>	<b>51</b>
<b>4.4.</b>	<b>Algoritmo de alineamiento.....</b>	<b>54</b>
<b>4.5.</b>	<b>Corrección de la distorsión radial .....</b>	<b>57</b>
4.5.1.	Modelo de distorsión radial .....	58
4.5.2.	Obtención de los parámetros de la distorsión radial .....	59
4.5.3.	Implementación del algoritmo de modelado y corrección de la distorsión radial.....	60
4.5.4.	Resultados, conclusiones y trabajo futuro .....	66
<b>4.6.</b>	<b>Ajuste de perspectiva.....</b>	<b>69</b>
4.6.1.	La transformación del plano proyectivo .....	70
4.6.2.	DLT normalizado para el cálculo de homografías 2D.....	71
4.6.3.	Implementación del algoritmo para conseguir el aumento de la resolución temporal .....	72
4.6.4.	Implementación del algoritmo para conseguir el aumento de la resolución espacial .....	76
4.6.5.	Conclusiones y trabajo futuro .....	78
<b>4.7.</b>	<b>Ajuste de contraste.....</b>	<b>79</b>
4.7.1.	Técnicas de ajuste de contraste.....	80
4.7.2.	Ajuste de contraste para el aumento de la resolución temporal .....	80
4.7.3.	Ajuste de contraste para el aumento de la resolución espacial .....	82
4.7.4.	Conclusiones.....	83
<b>4.8.</b>	<b>Resultados y conclusiones .....</b>	<b>84</b>
<b>5.</b>	<b><i>Conclusiones.....</i></b>	<b>86</b>
	<b>Trabajo futuro.....</b>	<b>87</b>
	<b><i>Referencias.....</i></b>	<b>87</b>
<b>A.</b>	<b><i>Manual de Usuario.....</i></b>	<b>88</b>
<b>A.1.</b>	<b>Introducción .....</b>	<b>88</b>
<b>A.2.</b>	<b>Ventana de inicio.....</b>	<b>88</b>
<b>A.3.</b>	<b>Modo AumentoTasa: Asistente de alineamiento y ajuste.....</b>	<b>89</b>
A.3.1.	Menú de Opciones .....	90
A.3.2.	Ventanas de visualización.....	90
A.3.3.	Cuadro de control.....	91
	HARDWARE PARAMETERS.....	91
	SCENE PARAMETERS .....	92
	ADVANCED PARAMETERS .....	92

CALIBRATION.....	94
SEQUENCE.....	94
ACQUISITION INFORMATION.....	94
<b>A.4. Modo AumentoTasa: Herramienta de captura y montaje.....</b>	<b>95</b>
A.4.1. Menú de opciones.....	96
A.4.2. Ventana de visualización.....	96
A.4.3. Cuadros de control.....	96
CAPTURE OPTIONS.....	97
RECORD.....	98
POST-RECORD CONTROLS.....	98
PANEL CONTROLS.....	99
CAPTURE OPTIONS.....	99
<b>A.5. Modo Panorámico: Asistente de alineamiento y ajuste.....</b>	<b>100</b>
CONTROLES COMUNES A AumentoTasa.....	101
HARDWARE PARAMETERS.....	102
CALIBRATION.....	102
<b>A.6. Modo AumentoCampo: Herramienta de captura y montaje.....</b>	<b>103</b>
<b>A.7. Cinematic Analysis.....</b>	<b>104</b>
A.7.1. Menú de Opciones.....	105
A.7.2. Ventana de visualización.....	105
A.7.3. Cuadros de control.....	106
SEQUENCE CONTROLS.....	107
REFERENCE INSERTION.....	108
DATA GENERATION.....	109
OVERVIEW.....	110
<b><i>PRESUPUESTO.....</i></b>	<b><i>111</i></b>
<b><i>PLIEGO DE CONDICIONES.....</i></b>	<b><i>112</i></b>

## ***INDICE DE FIGURAS***



<i>Ilustración 1 : Esquema ciclo de captura</i> .....	3
<i>Ilustración 2 : Cronograma de un ciclo de captura para el modo AumentoTasa</i> .....	4
<i>Ilustración 3 : MotionPro X Timing Hub</i> .....	5
<i>Ilustración 4 : Disposición de las cámaras modo AumentoCampo, juntos con el ejes de referencia</i> .....	6
<i>Ilustración 5 : Cámaras Motion Pro X4</i> .....	7
<b><i>Ilustración 7 : Cámaras en modo AumentoTasa</i></b> .....	8
<b><i>Ilustración 8 : lentes utilizadas</i></b> .....	8
<i>Ilustración 9 : Detalle MotionPro X Timing Hub</i> .....	9
<i>Ilustración 10 : Cámaras en modo AumentoCampo</i> .....	11
<i>Ilustración 11 : Interfaz de MATLAB 7.0.1 y toolbox GUIDE</i> .....	16
<i>Ilustración 12 : Jerarquía de objetos utilizados por MATLAB para la implementación de interfaces</i> .....	17
<i>Ilustración 13 : Aspecto general del interfaz de ajuste y alineamiento para el modo AumentoTasa</i> .....	21
<i>Ilustración 14 : Aspecto general del interfaz de captura y montaje</i> .....	22
<i>Ilustración 17 : Aspecto general de la interfaz de alineamiento y ajuste para el modo AumentoCampo.</i> ..	23
<i>Ilustración 18 : AumentoCampo , interfaz de captura y montaje . Pan_adquisicion.fig</i> .....	24
<i>Ilustración 19 : Aspecto general de la interfaz de Cinematic Analisis</i> .....	25
<i>Ilustración 20 : Interfaz grafica desarrollada con GUIDE y jerarquía de objetos</i> .....	26
<i>Ilustración 21 : Flujo de ejecución para las aplicaciones en modo AumentoTasa y AumentoCampo.</i> .....	27
<i>Ilustración 22 : Flujo de ejecución de la aplicación Analisis Cinematic</i> .....	28
<i>Ilustración 23 : Interfaz de Alineamiento y ajuste del modo AumentoTasa : US_Calibration.fig</i> .....	31
<i>Ilustración 24 : Interfaz de Captura y montaje del modo AumentoTasa : US_Acquisition.fig</i> .....	34
<i>Ilustración 25 : Interfaz de alineamiento y ajuste del modo AumentoCampo: Pan_Calibration.fig</i> .....	37
<i>Ilustración 26:Interfaz de Captura y montaje del modo AumentoCampo : Pan_Acquisition.fig</i> .....	39
<i>Ilustración 27 : Interfaz de Cinematic Analisis: Fase3.fig</i> .....	42
<i>Ilustración 28: Modelo pinhole</i> .....	47
<i>Ilustración 29: Esquema de la composición de la secuencia para el modo AumentoTasa</i> .....	49
<i>Ilustración 30: Esquema de las mejoras propuesta para mejorar la secuencia final en el modo US</i> .....	50
<i>Ilustración 31 : Esquema de ubicación de las cámaras para el modo AumentoCampo</i> .....	52
<b><i>Ilustración 32 : Esquema inicial para el aumento de la resolución espacial</i></b> .....	52
<b><i>Ilustración 33 : Diagrama de bloques para la obtención de una secuencia mejorada para el modo AumentoCampo</i></b> .....	53
<i>Ilustración 34 : Región común a imagenes capturadas en el modo AumentoTasa</i> .....	54
<i>Ilustración 35 : Formula utilizada para el cálculo del centro de masas</i> .....	55
<i>Ilustración 36 : Calculo del origen y dimensiones de la región común a dos imágenes</i> .....	56
<i>Ilustración 37 : Resultado del algoritmo de alineamiento</i> .....	56
<i>Ilustración 38 : Algunos tipos de distorsión interna</i> .....	57
<i>Ilustración 39 : Algunos tipos de distorsión externa</i> .....	57
<i>Ilustración 40 : Imágenes original, con distorsión barrel y pin-cushion</i> .....	58
<i>Ilustración 41 : Estructura utilizada para el cálculo de los parámetros de distorsión radial</i> .....	59
<i>Ilustración 42 : Ejemplo de imagen capturada para el algoritmo de corrección radial</i> .....	60
<i>Ilustración 43 : Etapa de detección de puntos característicos: imagen filtrada (arriba izda.), imagen filtrada y umbralizada junto con la imagen original (arriba dcha.), e imagen original con los puntos característicos localizados (abajo)</i> .....	62
<i>Ilustración 44 : Iteración para la estimación del primer coeficiente, <math>k_1</math>, que modela la distorsión radial</i> .....	64
<i>Ilustración 45 : Comando en Matlab que realiza una interpolación de segundo orden, interp2(Z,XI,YI)</i> . 65	65
<i>Ilustración 46 : Resumen del algoritmo que modela y corrige la distorsión radial</i> .....	66
<i>Ilustración 47: Aproximación inicial a la estimación de los coeficientes de la distorsión radial.</i> .....	67
<i>Ilustración 48: Resultado del algoritmo para 50 imágenes: mejora del error de linealidad (izda.), y valores obtenidos para el primer coeficiente (dcha.)</i> .....	68
<i>Ilustración 49: Resultados para una lente de 4.5 mm: imagen capturada (izda.) e imagen corregida (dcha.)</i> .....	68
<i>Ilustración 50 : Interfaz para la marcación manual de puntos y sus correspondientes</i> .....	74
<i>Ilustración 51: Resultado y detalle de la transformación del plano proyectivo de una imagen, utilizando homografías pero sin corregir la distorsión radial.</i> .....	75
<i>Ilustración 52 : Imagen panorámica compuesta utilizando 2 imágenes y una única homografía. Se ha ajustado el contraste .</i> .....	78
<i>Ilustración 53 : Esquema de ajuste de contraste para el modo UlstraSpeed</i> .....	82
<i>Ilustración 54 : Ventana de Inicio</i> .....	89

<i>Ilustración 55 : Aspecto general del interfaz de ajuste y alineamiento para el modo AumentoTasa</i> .....	89
<i>Ilustración 56 : Detalle menú de opciones</i> .....	90
<i>Ilustración 57 : detalle de la interfaz de alineamiento y ajuste: marcos de imágenes</i> .....	90
<i>Ilustración 58 : detalle de la interfaz de alineamiento y ajuste: cuadros de control</i> .....	91
<i>Ilustración 59 : detalle del interfaz de alineamiento y ajuste: Scene Parameters</i> .....	92
<i>Ilustración 60 : detalle del interfaz de alineamiento y ajuste: Advanced Parameters</i> .....	93
<i>Ilustración 61 : detalle del interfaz de alineamiento y ajuste</i> .....	94
<i>Ilustración 62 : Aspecto general del interfaz de captura y montaje</i> .....	95
<i>Ilustración 63 : detalle de la interfaz de captura y montaje: marco.</i> .....	96
<i>Ilustración 64: detalle de la interfaz de captura y montaje: cuadros de control</i> .....	97
<i>Ilustración 65 : AumentoTasa , ejemplo de ciclo de grabación para 3 cámaras (a)</i> .....	99
<i>Ilustración 66 : AumentoTasa ejemplo de ciclo de grabación para 3 cámaras (b)</i> .....	99
<i>Ilustración 67 : Detalle de la interfaz de captura y montaje: Capture controls</i> .....	100
<i>Ilustración 68 : Aspecto general del interfaz de alineamiento y ajuste para el modo AumentoCampo. .</i>	101
<i>Ilustración 69 : Detalle interfaz alineamiento y ajuste para el modo AumentoTasa , Hardware parameters</i> .....	102
<i>Ilustración 70 : Ventana emergente cuando se pulsa en botón sketch que indica la separación necesaria para componer la panorámica utilizada en el modo AumentoCampo</i> .....	103
<i>Ilustración 71 : AumentoCampo , interfaz de captura y montaje . Pan_adquisicion.fig</i> .....	103
<i>Ilustración 72 : Aspecto general de la interfaz de Cinematic Analisis</i> .....	104
<i>Ilustración 73: Menú de opciones de la interfaz de Cinematic Analisis</i> .....	105
<i>Ilustración 74 : Detalle Interfaz de Cinematic Analisis , ventana de visualización</i> .....	106
<i>Ilustración 75 : Detalle Interfaz de Cinematic Analisis , cuadros de control.</i> .....	107
<i>Ilustración 76 : Detalles de la interfaz de Cinematic Analisis)Opciones con los objetos</i> .....	108

## **INDICE DE TABLAS**

<i>Tabla 1 : Modelos de cámaras MotionPro y tasas de grabación</i> .....	8
<i>Tabla 2: Sistema de archivos para el modo AumentoTasa</i> .....	30
<i>Tabla 3 : Métodos implementados en el archivo US_Calibration.m</i> .....	32
<i>Tabla 4 : Métodos implementados en el archivo US_Acquisicion.m</i> .....	34
<i>Tabla 5 : Sistema de archivos para el modo AumentoCampo</i> .....	36
<i>Tabla 6 Métodos implementados en el archivo Pan_Calibration.m</i> .....	37
<i>Tabla 7: Métodos implementados en el archivo Pan_Acquisicion.m</i> .....	39
<i>Tabla 8 : Sistema de archivos para la aplicación de cálculo de medidas, Cinematic Analisis</i> .....	40
<i>Tabla 9 :Métodos implementados en el archivo Fase3.m</i> .....	42
<i>Tabla 10 : Tiempo de ejecución del algoritmo de corrección radial</i> .....	68
<i>Tabla 11 : Binning , resolución y modelo de cámara</i> .....	93
<i>Tabla 12 : Ganancia de píxel</i> .....	93

# 1. Introducción

---

## 1.1. Motivación

Los sistemas y cámaras de alta velocidad se emplean en multitud de aplicaciones industriales, científicas, aeronáuticas, etc. Entre las aplicaciones más destacadas se encuentran: pruebas de choque en automoción, balística, control de fabricación, estudios en dinámica de fluidos, control de turbulencias, visualización de explosiones, biomecánica, anuncios de publicidad, formula 1, reacciones químicas...

El objetivo del proyecto es crear una serie de herramientas y protocolos de apoyo para un sistema de captación de secuencias de vídeo con múltiples cámaras de alta velocidad. Como resultado, se pretende generar una única secuencia que verá mejorada su resolución temporal (aumento del número de cuadros por segundo al situar varias cámaras desfasadas en el tiempo apuntando a un mismo objetivo) o espacial (aumento del campo de visión horizontal proporcional al número de cámaras que se sitúan paralelas apuntando a objetivos contiguos).

Aparte de la puesta en marcha del sistema, en ambos casos son necesarias herramientas que, antes de realizar la adquisición de imágenes, faciliten la correcta colocación de las cámaras y ayuden a la calibración de los parámetros internos de las mismas. Asimismo, una vez realizada la captura, se requieren herramientas que corrijan deformaciones y desviaciones introducidas por las limitaciones asociadas al propio diseño del sistema.

## 1.2. Objetivo

Para el desarrollo de este PFC, se considera un sistema de hasta cuatro cámaras de alta velocidad (Motion Pro X) en dos posibles escenarios:

Escenario 1: las cámaras se colocan con sus ejes paralelos, lo más compactadas que resulte posible y ancladas en una estructura fija (dos arriba, dos abajo) de modo que todas enfoquen a un mismo objetivo. Se activan sus entradas de sincronismo de modo que capturen desfasadas unas de otras. Como resultado de la intercalación de las secuencias captadas por cada cámara se obtiene una única secuencia cuya resolución temporal (cuadros por segundo o *frames per second – fps* -) se verá aumentada por cuatro, a costa de perder parte del campo de visión global de cada cámara. Este es el denominado escenario o modo *AumentoTasa*.

Escenario 2: las cámaras se colocan con sus ejes paralelos sobre unos trípodes alineados perpendicularmente a dichos ejes, situadas a la misma altura y separadas una distancia fija; todas las cámaras apuntan, por tanto, en una misma dirección perpendicular a la de alineamiento. Se activan sus entradas de sincronismo de modo que capturen en sincronía. Como resultado de la fusión de las secuencias captadas por cada cámara se obtiene en este caso una única secuencia correspondiente a un campo de visión espacial (dimensión horizontal) hasta cuatro veces el campo de visión horizontal de una de las cámaras. Este es el denominado escenario o modo *AumentoCampo*.

En el primer escenario, antes de realizar la adquisición de las imágenes es necesario que las imágenes captadas por cada una de las cámaras sean lo mas parecidas posible. Por ello será necesario implementar un algoritmo que calculen la ROI (*Region of Interest*) o región espacial que debe capturar cada cámara y ajustar ciertos parámetros de las cámaras (sensibilidad, tiempo exposición, etc.) basándose en unos parámetros, conocidos a priori, relativos a la escena que desea captar.

En el segundo escenario, antes de realizar la adquisición de las imágenes es necesario que las cámaras estén bien situadas para conseguir el efecto visual deseado. Será necesario implementar unos algoritmos que ayuden a colocar y a ajustar de manera dinámica las cámaras en su posición correcta y que calibren los parámetros de las cámaras en función, de nuevo, de unos parámetro relativos a le escena que se desea captar.

Durante la fase de captura de imágenes, es necesario que las cámaras estén sincronizadas correctamente. Para el primer escenario la sincronización consistirá en un desfase temporal, de modo que las imágenes correspondientes a cada cámara se encuentren intercaladas en el tiempo a intervalos regulares. Para el segundo escenario, la captura deberá ser simultánea en todas las cámaras.

Para la composición de la secuencia final, será necesaria una serie de algoritmos que mejoren la calidad de la secuencia corrigiendo las aberraciones y deformaciones introducidas por la imperfección del sistema.

Para la comprobación de los algoritmos realizados se propone implementar una aplicación programada con Matlab para calibrar las cámaras, capturar las secuencias y presentar la secuencia resultante en ambos escenarios. Además se ha implementado una aplicación, en el mismo entorno de programación, para realizar medidas físicas de los objetos que aparecen en la secuencia resultante final o cualquier otra secuencia.

### **1.3. Estructura de la memoria**

El documento que describe este Proyecto Fin de Carrera se organiza de la siguiente manera:

- Capitulo 2: se presenta el sistema, componentes y montaje , y los fundamentos teóricos en la que este proyecto ha sido basado para conseguir los objetivos expuestos anteriormente
- Capitulo 3: se presenta una explicación detallada de las aplicaciones implementadas para la configuración y control del sistema en cada modo. Teniendo en cuenta que el sistema trabaja con cámaras de la empresa Red-Lake . En la primera parte de este capitulo se explica la funcionalidad de la aplicación, una especie de manual para el usuario. En la segunda parte, orientada mas a conocer como fue desarrollada, se exponen el sistema de archivos , métodos implementados variables utilizadas mas importantes así como unos diagramas de flujo de cada una de las aplicaciones implementada.

- Capitulo 4: en este capitulo se analizan los principales defectos de las secuencias resultantes de cada uno de los modos. Se analizan cuales son mas determinantes, se proponen varios algoritmos y se implementan adaptándose a cada uno de los modos de manera que la calidad en la secuencia final sea mejorada.

## 2. Antecedentes y planteamiento

---

Una cámara normal, captura y reproduce imágenes a una tasa de 24 o 25 *fps* (*frames per second* o imágenes por segundo), la televisión reproduce a 25 *fps* (PAL). Las cámaras de alta velocidad pueden capturar a tasas superiores a 250 000 *fps*, basándose en el principio de Miller , Charles Miller 1930 . La idea es que si se forma una imagen en un plano de un espejo que rota, se puede cambiar el ángulo de este espejo sin apreciar en la imagen captada ninguna apariencia de movimiento debido a esta rotación. La primera cámara con estas características fue utilizada en el proyecto Manhattan, para analizar la detonación de los primeros dispositivos nucleares. En los años 50, esta tecnología fue explorada por el sector privado y empezó así su comercialización y desarrollo. Hoy en día la técnica de espejo rotatorio es la única manera para capturar a altas tasas.

Actualmente la película ha sido sustituida por sensores CCD pero el modo de capturar a altas tasas no ha cambiado. Todos los fabricantes de cámaras de alta velocidad, están centrando sus esfuerzos en dispositivos digitales que generan más beneficios y posibilidades de operación que las cámaras de película convencionales.

Un problema de estas cámaras es la necesidad de exposición de la película o sensor. A altas tasas, se necesita una sobreiluminación de la escena para poder observar el fenómeno capturado. Algunas veces, esta iluminación genera tal cantidad de calor que destruye el objeto en análisis.

Actualmente, recientes estudios sobre la formación de imágenes han mejorado la resolución temporal a menos de 50 *ps* , lo que supone una tasa de grabación superior a 20 000 000 000 *fps*. Estos dispositivos operan transformando la luz incidente (fotones) en un flujo de electrones que son deflectados en un fotoánodo que se plasma en una película o sensor CCD. La calidad (rango dinámico y sensibilidad a la iluminación) de la imagen limita las aplicaciones de este tipo de cámaras. En enero de 2008 la ‘mejor’ cámara de alta velocidad disponible era la Phantom HD capaz de capturar 1000 *fps* con una resolución de 1920 x 1080 con una profundidad de 14 bits a color.

La utilización de estas tasas de grabación queda ya fuera del ámbito recreativo, las películas denominadas *slow-motion* utilizan tasas de 500 *fps*. La principal utilidad de este tipo de secuencia, a tasas superiores a los 1000 *fps*, es el estudio de los fenómenos que en ella ocurren. Las imágenes capturadas son analizadas para determinar algunos parámetros físicos (velocidad, trayectoria, aceleración...) de los objetos que aparecen . Es por este motivo por lo que se utilizan sensores digitales, CCD, para capturar las imágenes y así generarlas en formato digital donde son más precisos y sencillos estos estudios.

Las cámaras utilizadas en este trabajo, Modelo Y-3 , capturan a una tasa de 1035 fps a una resolución de 1280 x 1024 con una profundidad de 8 bits a color. Las aplicaciones y herramientas implementadas permiten, combinando 4 cámaras, obtener una secuencia con una resolución de 1280 x 1024 capturada a una tasa de 4140 fps. Además, con el modo de operación *AumentoCampo* , se consigue una secuencia capturada a 1035 fps con una resolución aproximada de 4736 x 1024. Estas características actualmente ninguna cámara en el mercado las iguala ni se aproxima. Estas tasas de grabación y resolución se alcanzan suponiendo una iluminación de la escena ideal.

El objeto de este proyecto es desarrollar unas herramientas para el aumento de la resolución tanto espacial como temporal. Estas herramientas han sido desarrolladas basándose en un sistema que se describe a continuación.

El sistema se basa en un grupo de cámaras, mínimo dos, máximo cuatro, con mismo modelo y características (idéntica óptica, tiempo de exposición...). La sincronización temporal y la ubicación de las cámaras esta determinada por el objetivo deseado.

A continuación se redactan los fundamentos para conseguir estas aspiraciones; además se describen las cámaras y dispositivos utilizados para establecer el sistema así como su montaje.

### **2.1. Planteamiento del objetivo**

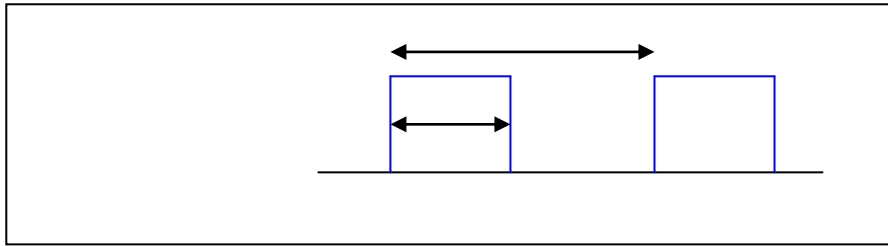
#### **2.1.1. Fundamentos del aumento de la resolución temporal**

Para conseguir el aumento de la resolución temporal, todas las cámaras captan un mismo objeto desde un mismo punto de vista (en la medida de lo posible) Las cámaras están desfasadas en el tiempo de modo que se multiplica el número de imágenes del objeto captadas por segundo por el número de cámaras.

Para comprender esta idea se define los siguientes parámetros relacionados con secuencias de imágenes que se pueden visualizar en la Ilustración 1:

- Tiempo de exposición ( $t_{exp} < 1/fps$ ): hace referencia al tiempo que el obturador permanece abierto, de modo que la película o el sensor (según si se trata de fotografía analógica o digital) recibe luz para la composición de la fotografía.
- El tiempo de exposición viene determinado por el periodo de adquisición. Un tiempo menor al necesario producirá una fotografía oscura (subexpuesta) y un tiempo mayor al requerido producirá una fotografía más clara (sobreexpuesta).
- El tiempo óptimo de exposición, dependerá de la luminosidad de la escena a fotografiar, de la apertura del diafragma y de la sensibilidad de la película o sensor.

- Periodo de adquisición ( $T = 1 / fps$ ): a la velocidad a la que se abre y cierra el obturador. tiempo entre capturas consecutivas. Éste es mayor que el tiempo de exposición ya que es necesario un tiempo extra para transferir las imágenes desde el sensor a la memoria de la cámara.
- Cuadros por segundo (*frames per second - fps*): es una medida de la frecuencia a la que un dispositivo produce únicas y consecutivas imágenes. Indica el número de imágenes por segundo de la cámara o de la secuencia resultante.



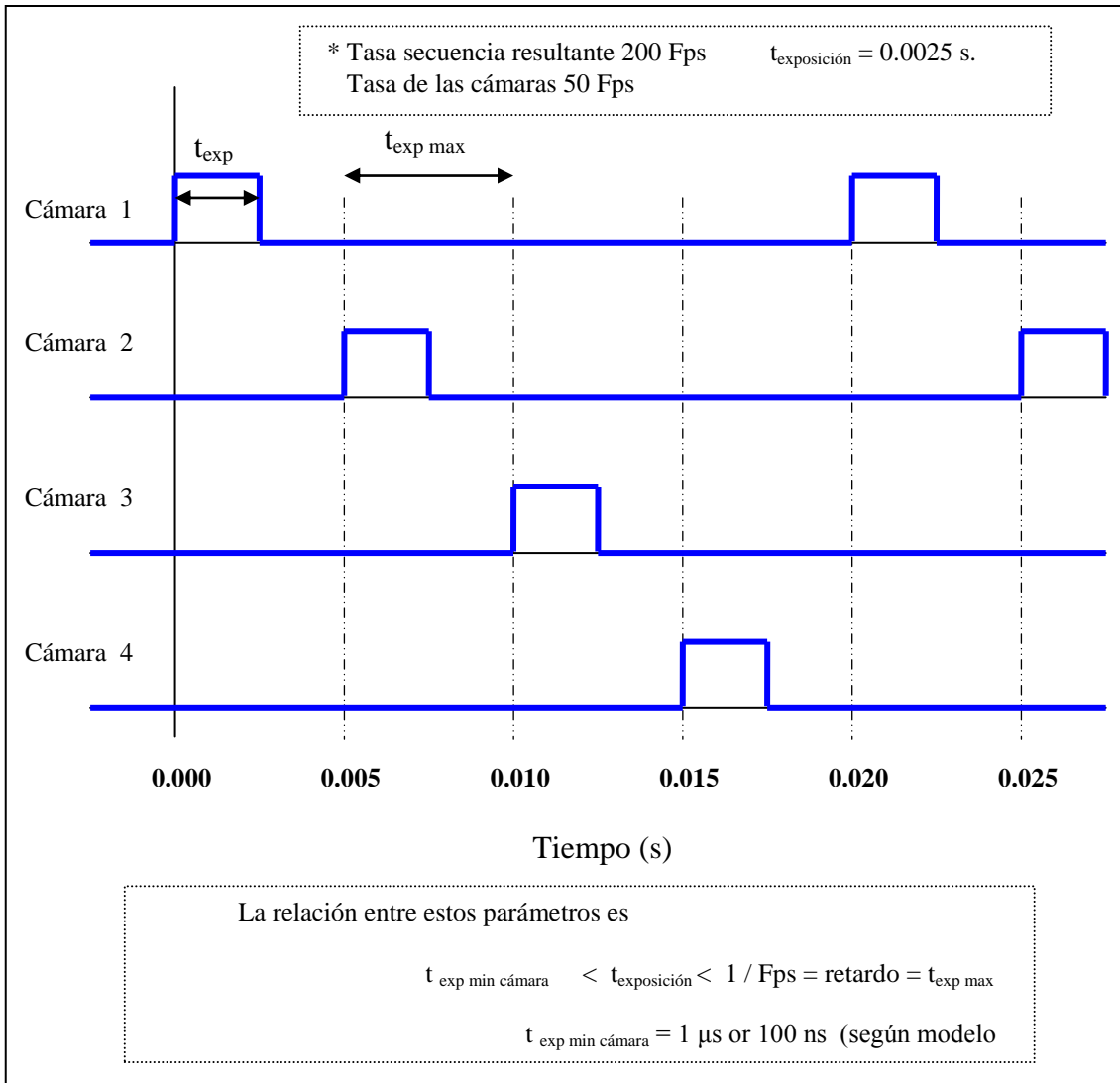
**Ilustración 1 : Esquema ciclo de captura**

Para conseguir un aumento de la resolución temporal, el tiempo en el que el obturador está cerrado para una de las cámaras ( $T - t_{exp}$ ) es utilizado por el resto para realizar sus capturas sin que haya solape temporal entre ellas. Esto obliga a que sea necesario que se cumpla:

$$T > \text{numero de cámaras (N)} * t_{exp}.$$

, donde N es el número de cámaras.

La siguiente figura, Ilustración 2, ilustra de manera práctica los conceptos definidos:



**Ilustración 2 : Cronograma de un ciclo de captura para el modo AumentoTasa**

### 2.1.2. Problemática asociada al proceso de aumento de la resolución temporal

Debido a que cada parte de la secuencia final está captada por una cámara distinta, se apreciarán diferencias de contraste, iluminación, enfoque, etc., bien entre cuadros consecutivos.

La imposibilidad de situar todas las cámaras en un mismo punto hace que las secuencias captadas no reflejen exactamente la misma escena, al variar el punto de vista. El efecto será tanto mayor cuanto más separadas estén las cámaras relativamente a la distancia que las separa del objeto que captan y cuanto mayores sean las deformaciones introducidas por sus ópticas.

Debido a esta situación, es necesario aplicar a la secuencia resultante una serie de algoritmos que mejoren la calidad. Previo a la captura, es recomendable situar las cámaras de manera óptima para minimizar el error de perspectiva y realizar un cálculo de la ROI (*Region of interest*) común a las cámaras. En este sentido se ha optado por



situar sus ejes paralelos y lo más juntos posible, anclando las cámaras sobre una estructura fija.

Una vez realizadas las capturas por cada una de las cámaras, al proceso de fusión (intercalar imágenes de cada una de las cámaras) se le añade un proceso de mejora de la calidad donde se hacen ajustes de perspectiva, corrección de la distorsión radial y ajuste de contraste.

### Sincronización

Aparte de los problemas mencionados, la sincronización resulta fundamental para la obtención del objetivo. Esta sincronización se consigue mediante un dispositivo *hardware*, un *timing hub* (Ilustración 3) que genera las señales necesarias con la precisión que necesita el sistema, ya que las cámaras utilizadas permiten altas tasa de grabación. Un ejemplo de las señales que tiene que generar este dispositivo se muestra en la figura anterior.

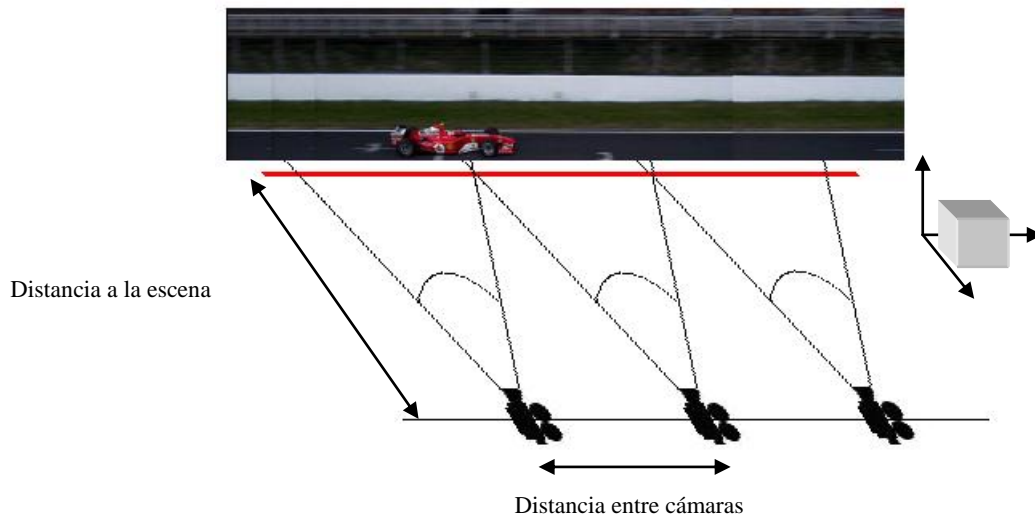
Además una señal debe indicar el comienzo de la grabación, el trigger. Esta señal debe llegar de manera simultánea a todas las cámaras.



Ilustración 3 : MotionPro X Timing Hub

### 2.1.3. Fundamentos del aumento de la resolución espacial

Como resultado de este objetivo, se obtiene una secuencia imagen panorámica compuesta a partir de las secuencias individuales capturadas por cada una de las cámaras. Las cámaras se sitúan a lo largo de un eje apuntando en dirección perpendicular a dicho eje y operando en sincronía, ver Ilustración 4. Su situación es tal que la imagen captada por cada una se solapa mínimamente con la captada por sus vecinas, de modo que es posible generar una secuencia de imágenes cuyo campo de visión es tantas veces más ancho como cámaras intervengan.



**Ilustración 4 : Disposición de las cámaras modo AumentoCampo, juntos con el ejes de referencia**

### **2.1.4. Problemática asociada al proceso de aumento de la resolución espacial**

Al igual que le caso anterior, debido a que cada parte de la secuencia final está captada por una cámara distinta, se apreciarán diferencias de contraste, iluminación, enfoque, etc., entre las distintas regiones solapadas de la secuencia final.

Otra fuente de error apreciada es la posible deformación introducida por la óptica utilizada. Además la dificultad a la hora de conseguir que la secuencia panorámica resultante sea coherente tanto espacial como temporalmente:

La coherencia espacial, trata de buscar un alineamiento (horizontal y vertical) en los planos comunes (zonas donde se produce solape) para conseguir continuidad en el plano de la escena. También, es necesario conocer *a priori* las características de la escena (distancia al objeto de grabación, tipo de lente...), ya que la separación entre las cámaras dependerá de estos valores. La coherencia temporal se trata en el apartado de sincronización.

Por lo tanto, la calidad del proceso final de fusión de imágenes dependerá de la situación relativa de las cámaras y de sus ópticas, de la uniformidad de los ajustes y parámetros de todas las cámaras, y de las técnicas de proceso de imagen aplicadas para corregir los defectos introducidos.

## Sincronización

Para conseguir la sincronización de la grabación para las cámaras, es necesario:

- Una señal para indicar el comienzo de grabación para todas las cámaras. Para este sistema se utiliza un trigger manual ya que el SDK no permite un trigger simultáneo a todas las cámaras ejecutado desde la aplicación, trigger software.
- Una señal común a todas las cámaras que indique en que instante debe capturar la imagen y durante cuanto tiempo. Ésta señal de sincronismo la puede aportar el *timing hub* utilizado en el modo *AumentoTasa* o bien una cámara denominada *Master*.

## 2.2. Arquitectura del sistema

A continuación se detallaran los componentes del sistema, diseñado para desarrollar una aplicación que hiciera posible conseguir los objetivos de este proyecto.

### 2.2.1. Cámaras



**Ilustración 5 : Cámaras Motion Pro X4**

El sistema consta de hasta cuatro cámaras de alta velocidad, modelos MotionPro® X producidos por la empresa RedLake con las siguientes características:

- Cámaras con sensores CMOS que pueden funcionar en modo doble exposición ,
- Interfaz Gigabit Ethernet y USB 2.0.
- 4 GB de memoria interna con descarga de alta velocidad.
- Resoluciones y tasa de grabación ( Fps ) :

**Tabla 1 : Modelos de cámaras MotionPro y tasas de grabación**

MotionPro X Camera		
Model	Resolution	Max Rate [fps]
X3	1280H x 1024V	1,040
X4	512H x 512V	5,100
X5	2352H x 1728V	250
MotionPro Y Camera		
Model	Resolution	Max Rate [fps]
Y3	1280H x 1024V	1,040
Y5	2352H x 1728V	500

- <http://www.redlake.com/products/MotionPro/default.aspx>

Las lentes utilizadas, marca NIKON, son de óptica fija: 12 mm, 28 mm, y 105 mm de distancia focal. Las dos primeras son adecuadas para capturar escenas cercanas a las cámaras (< 8 metros) y la de 105 mm para escenas lejanas a las cámaras (> 20 metros), aunque la elección de la lente mas adecuada dependerá de la escena que se desee captar.

Las cámaras se conectan (mediante cable USB o Ethernet) al ordenador donde la aplicación desarrollada permite configurarlas, capturar secuencias y descargarla a disco.

Antes de continuar es necesario separar los dos objetivos del proyecto: el *aumento de la resolución temporal (AumentoTasa)* y *aumento de la resolución espacial (Panorámico)*.

### 2.2.2. Sistema *AumentoTasa*: elementos y montaje

Los elementos necesarios para montar el sistema en este modo son:



**Ilustración 6 : Cámaras en modo AumentoTasa**



**Ilustración 7 : lentes utilizadas**

- 2 o 4 cámaras, el sistema está diseñado para operar con hasta 4 cámaras RedLake MotionPro HS (en resoluciones de 1280x1024 o de 512x512).
- Ópticas fijas de diversas distancias focales a elegir con sus adaptadores, ver Ilustración 7.
- Placas para crear la estructura donde se sustentaran las cámaras. Las cámaras se colocan lo más juntas posibles. Estas placas fijan las cámaras para evitar que se muevan y crear una estructura estable. Ver Ilustración 6
- Trípode para soportar la estructura con las cámaras.
- MotionPro X Timing Hub, aporta hasta 9 señales de sincronización con 20 ns de resolución para sincronizar las cámaras. Para el sistema tan solo se utilizan 4 señales, desde la 0 hasta la 3, Ilustración 8.



**Ilustración 8 : Detalle MotionPro X Timing Hub**

- 2 o 4 cables coaxiales con conectores BNC macho, para la interconexión de las cámaras con el *timing hub*. Se debe conectar a las salidas del *timing hub*, empezando desde el 0 con el conector hembra SYNC IN de cada cámara.
- Sistema de disparo, para un sistema con dos cámaras necesitamos dos cables coaxiales unidos en un conector en T, en la tercera patilla del conector irá conectado el disparador. Los dos cables irán conectados a la entrada de TRIG IN de cada una de las cámaras, de modo que cuando activemos el pulsador se ilumine el *led* de TRIG IN de todas las cámaras. Este sistema es necesario para indicar a las cámaras el comienzo de la grabación.
- Si el ordenador no dispone del número necesario de puertos USB 2.0, es necesario un *hub* USB para conectar las cámaras y el *timing hub* con el ordenador. Este dispositivo es necesario alimentarlo mediante corriente eléctrica.
- Cables especiales para conectar las cámaras con el ordenador. Los suministra el fabricante de las cámaras; tienen en un extremo un conector USB 2.0 y en el

otro extremo una conexión especial para la cámaras. También es posible conectarlas utilizando cables *Ethernet*.

- Cables de corriente con transformadores, para alimentar las cámaras y el *timing hub*. Son suministrados por el fabricante de éstos dispositivos. Se recomienda utilizar un alargador con enrollador de cable.

- Iluminación: si se desea capturar a altas tasas es necesario iluminar la escena. Las técnicas de iluminación jugarán una importante baza en la estructura final de la escena. El análisis de este aspecto queda fuera del ámbito de este proyecto.

### 2.2.3. Sistema *AumentoCampo* : elementos y montaje

Para montar este sistema se añaden o eliminan los siguientes elementos con respecto al modo *AumentoTasa* :

- Se elimina el dispositivo de sincronización, esta sincronización se consigue eligiendo una cámara del sistema como *MASTER*. La cámara *MASTER* se configura para que capture utilizando una señal que genera de manera interna. Esta señal se transmite, a través de cables coaxiales, al resto de las cámaras. La señal de sincronismo sale de la cámaras por el conector denominado *SYNC OUT*, y puede recibir la señal de sincronismo por la entrada *SYNC IN* (debe estar configurada para ello).
- Para un sistema de dos cámaras, necesitamos un cable coaxial que conecte la salida de *SYNC OUT* de la cámara *MASTER*, con la entrada de *SYNC IN* de las otras cámaras.
- En un sistema de cuatro cámaras, se necesitan tres cables. Se conecta la salida de *SYNC OUT* de la cámara *MASTER*, con la entrada de *SYNC IN* de la cámara más cercana. De esta cámara, se conecta con otro cable la salida de *SYNC OUT* con la entrada de *SYCN IN* de la cámara siguiente. La señal de sincronismo le llega a la última cámara de la misma manera.
- También se necesitan dos o cuatro trípodes, según el número de cámaras. Estos trípodes tienen que estar alineados paralelamente al plano de la escena, también tienen que situar las cámaras a la misma altura y estar equilibrados a nivel. La separación entre las cámaras depende de la distancia del sistema a la escena y de la lente (distancia focal) utilizada. Ver Ilustración 9.



Ilustración 9 : Cámaras en modo AumentoCampo

### 2.3. Diseño software

Las cámaras y el *timing hub* solo pueden ser controlados y configurados mediante aplicaciones específicas, ya que no poseen software interno. Existe un paquete de aplicaciones que se distribuye junto con las cámaras, *Motion Pro X Studio* entre otras, con este fin. Estas aplicaciones no son lo suficientemente potentes para conseguir el objetivo de este Proyecto; además no consideran ni permiten el tratamiento algorítmico necesario previo, durante y posterior a la captura, por lo que se consideró necesario desarrollar una aplicación específica para este objetivo.

Una vez definido el sistema con el que se conseguirá el objetivo, se optó por implementar aplicaciones para la configuración y control de las cámaras y del dispositivo de sincronización, en caso de ser necesaria su utilización.

Además, para poder realizar medidas físicas sobre las imágenes y ponderar la calidad de la secuencia final se planteó una aplicación adicional. Ésta se ejecuta de manera independiente de las anteriores y permite obtener distintas medidas físicas del objeto de grabación basándose en una serie de puntos marcados de manera manual.

Tal y como se comentó anteriormente, la aplicación de control y configuración de las cámaras necesitaba incorporar una serie de algoritmos que mejoraran la calidad de la secuencia final y que minimizara los errores introducidos por la configuración del sistema. Estas herramientas están presentes durante el proceso de captura, posterior y anterior a la misma.

### 2.4. Conclusiones

Una cámara de alta es un dispositivo utilizado para grabar películas *slow-motion* o estudios de fenómenos transitorios que ocurren a muy alta velocidad. Estas cámaras capturan a altas tasas utilizando una aplicación del principio de Miller que se basa en un espejo o prima rotatorio que plasma en la película o sensor imágenes sin apreciación de movimiento. Actualmente, las películas utilizadas antiguamente han sido sustituidas por

sensores digitales CCD, que generan imágenes en formato digital con los beneficios (económicos y funcionales) que esto provoca.

Estas cámaras tienen que reducir la resolución para conseguir mayores tasas de grabación, o viceversa. El principal inconveniente es la sensibilidad de los sensores o películas utilizadas, a muy altas tasas el tiempo de exposición es muy pequeño y es necesaria una sobreiluminación de la escena para que en el sensor se plasme una imagen nítida sobre el sensor.

La utilidad de cámaras que capturen a muy altas tasas y/o con una resolución panorámica se limita a usos militares y científicos. Estas secuencias son analizadas para obtener medidas de magnitudes físicas (velocidad, trayectoria ...) acerca de los objetos que aparecen en ella.

El objeto de este proyecto es aumentar la tasa de grabación o resolución a partir de secuencias capturadas por cámaras individuales con el objetivo de realizar medidas sobre objetos que aparezcan en la escena. Para aumentar la tasa de grabación, las cámaras implicadas capturan en instantes distintos de manera la secuencia final se obtiene intercalando las imágenes de cada cámara. En este modo, las cámaras se colocan lo más próxima posible, de manera que capturen desde el 'mismo' punto de vista. En el segundo objetivo, las cámaras se ubican separadas una distancia determinada por la lente utilizada y la distancia de la escena a las cámaras. En este modo, las cámaras capturan en el mismo instante. La secuencia final se consigue fusionando las imágenes de cada cámara en cada instante para conseguir una imagen panorámica. Estas operaciones de intercalado y fusión, generan imperfecciones que es necesario corregir para generar secuencias que resulten útiles a la hora de ser analizadas.

Para desarrollar estas herramientas, son necesarias obtener secuencias capturadas según el modo de operación. En este proyecto, se utilizan cámaras de la empresa Red-Lake quienes aportan un software junto con ellas que no es lo suficientemente potente para conseguir los objetivos. Por este motivo fue necesario implementar una aplicación propia (para cada modo) para la configuración de los parámetros internos y ubicación de las cámaras implicadas. Además, para realizar medidas sobre las secuencias obtenidas, o cualquier otra que se desee analizar, se ha implementado una aplicación que se basa en la marcación manual de puntos para calcular distintas magnitudes físicas y los presenta en un formato sencillo para el usuario.



## 3. Desarrollo software

---

Para comprobar la utilidad y necesidad de las herramientas implementadas, según el objetivo de este Proyecto, se necesitan secuencias de partida capturadas por los sistemas descritos en el apartado anterior.

Para conseguir estas secuencias, además de montar el sistema , se necesita que las cámaras estén configuradas según el modo de operación .Esta configuración y control de las cámaras (y del *timing hub*) se realiza a través de software. Han sido desarrolladas varias aplicaciones propias para controlar y configurar las cámaras en cada modo. Además se ha desarrollada otra aplicación independiente para hacer análisis cinemáticos sobre secuencias.

En este apartado primero se explican las especificaciones iniciales que deben cumplir cada aplicación desarrollada: *AumentoTasa*, *AumentoCampo* y *CinematicAnalysis*

En el siguiente apartado, se explica la funcionalidad de las aplicaciones desarrolladas. Se trata de una especie de manual de usuario que describe la funcionalidad de las aplicaciones implementadas.

El último apartado describe el funcionamiento de las aplicaciones. Para cada una , se hace una descripción resumida de los archivos desarrollados, diagramas de flujos y variables utilizadas.

### 3.1. Especificaciones iniciales

Previo a la etapa de desarrollo, es necesario tener claro cuales son los objetivos de la aplicación que se va a implementa. En este apartado se exponen los requisitos iniciales sobre funcionalidad y características de la interfaces desarrolladas.

#### 3.1.1. Descripción y objetivos iniciales

El objetivo de este proyecto es crear una serie de herramientas y protocolos de apoyo para un sistema de captación de secuencias de vídeo con múltiples cámaras de alta velocidad.

El proyecto plantea dos situaciones bien diferenciadas, según se ha detallado en el Capítulo 1:

- Aumento de la resolución temporal de la secuencia (*AumentoTasa*)
- Aumento del plano de la secuencia (*AumentoCampo*)

Las herramientas que se desea desarrollar son:

- Un *asistente de alineamiento y ajuste* que ayude a posicionar adecuadamente las cámaras y ofrezca indicaciones sobre la relación entre estas posiciones y los posibles errores o aberraciones apreciados en la secuencia generada. Asimismo permitirá controlar parámetros básicos de las cámaras (sensibilidad, velocidad, etc.)
- Una *herramienta de captura y montaje* que genere, en cada una de las dos situaciones contempladas, la secuencia de vídeo resultante del proceso de fusión de las secuencias captadas con cada una de las cámaras.
- Una *herramienta de análisis* que permita, en cada una de las dos situaciones contempladas, situar marcas sobre cada cuadro de la secuencia (puntos, líneas, ejes de giro, etc.) y generar ficheros de datos (posiciones, velocidades, velocidades angulares, etc.) sobre el objeto filmado.

La calidad del proceso final de fusión de imágenes dependerá de la situación relativa de las cámaras y de sus ópticas, de la uniformidad de los ajustes y parámetros de todas las cámaras, y de las técnicas de proceso de imagen aplicadas para corregir los defectos introducidos.

### **3.1.2. Requisitos**

Las herramientas a desarrollar estarán integradas en una única aplicación que funcionará bajo MS Windows XP, con una interfaz gráfica típica. Sus requisitos particulares se exponen a continuación. Asimismo, en la sección se muestra una composición gráfica de la funcionalidad esperada de cada herramienta.

#### **3.1.2.1.ASISTENTE DE ALINEAMIENTO Y AJUSTE**

Ha de ofrecer un protocolo sencillo para guiar el posicionamiento de las cámaras en las dos situaciones descritas. El protocolo contará de un *ajuste inicial* basado en la especificación de parámetros básicos del entorno (tipo de lentes, distancia al objeto, velocidad aproximada, etc.) y en indicaciones textuales y gráficas de cómo situar las cámaras; y de un *ajuste fino* basado en la visualización en vivo de las imágenes captadas por las cámaras así como de indicaciones de las correcciones de posición necesarias. El asistente también permitirá ajustar parámetros básicos de captura de las cámaras. Esta etapa requiere el SDK de las cámaras.

#### **3.1.2.2.HERRAMIENTA DE CAPTURA Y MONTAJE**

Mediante la utilización del SDK de las cámaras, la herramienta lanzará el proceso de grabación. A continuación descargará parcialmente las series de ficheros de imagen procedentes de las cámaras con el fin de ofrecer la posibilidad de previsualizar partes de la secuencia resultante, a baja resolución y sin efectuar ningún tipo de proceso de imagen, para así validar el proceso de captación. Se ofrecerá la posibilidad de incluir en la secuencia final marcas temporales y un logotipo. En esta primera aproximación, el proceso de montaje consistirá en el intercalado (primera situación) o concatenado (segunda situación) de las imágenes de todas las cámaras. La aplicación de técnicas de post-proceso de imagen para mejorar la calidad de las secuencias obtenidas es el

objetivo de una fase posterior del proyecto, cuya especificación y desarrollo se detallará en el Capítulo 4

### **3.1.2.3.HERRAMIENTA DE ANÁLISIS**

El objetivo de esta herramienta es obtener estimaciones de los parámetros de movimiento del objeto filmado. Para ello, la herramienta permitirá situar sobre cada cuadro de la secuencia diversos tipos de marcas (puntos, líneas, ejes de giro, etc.). Una vez terminado el proceso de marcación, la herramienta realizará los cálculos necesarios y volcará los datos a un fichero que se pueda importar desde MS Excel con facilidad.

## **3.2. Herramienta de desarrollo: Matlab 7.0.1**

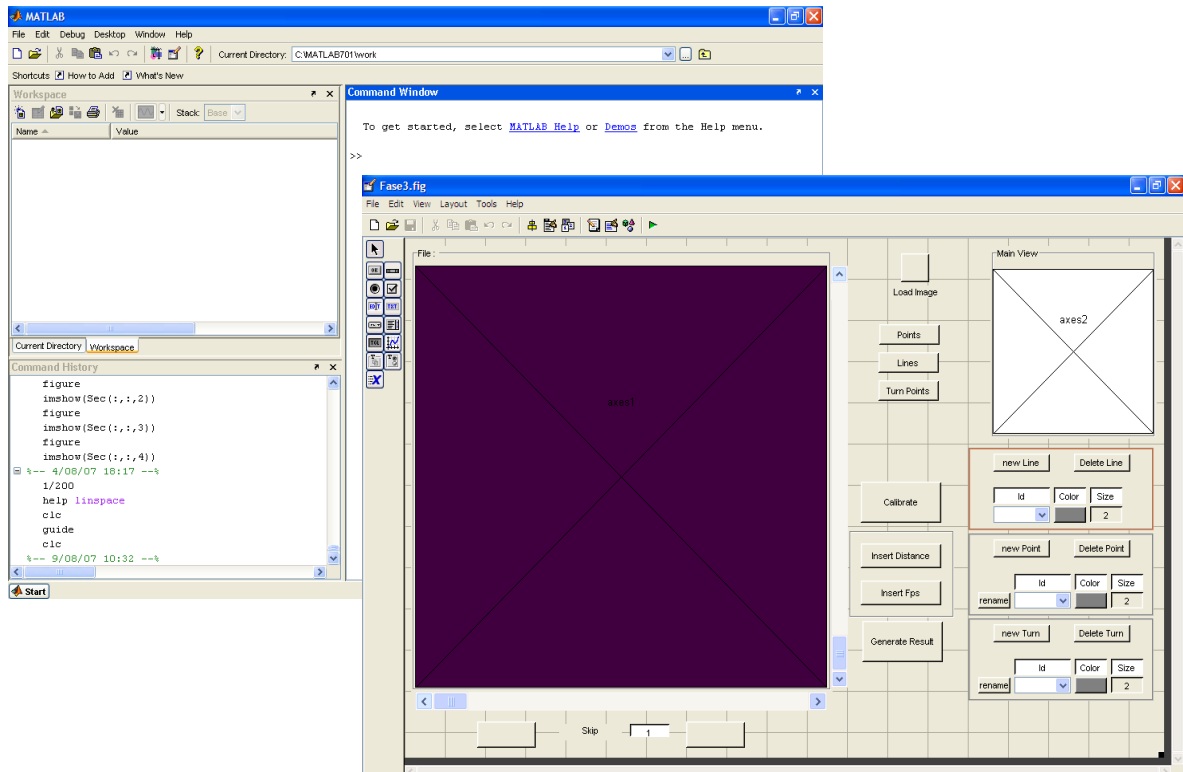
### **3.2.1. Entorno de desarrollo**

MATLAB es la abreviatura de *Matrix Laboratory* (laboratorio de matrices). Es un programa de análisis numérico creado por *The MathWorks* en 1984. Está disponible para las plataformas Unix y Windows.

Se pueden ampliar sus capacidades con *Toolboxes*; algunas de ellas están destinadas al procesado digital de señal, adquisición de datos, economía, inteligencia artificial, lógica difusa... También cuenta con otras herramientas como *Simulink*, que sirve para simular sistemas y un compilador propio para construir aplicaciones autoejecutables.

En los últimos años ha incluido muchas más funcionalidades, como la de programar directamente procesadores digitales de señal, crear código VHDL y otras. MATLAB es un programa de cálculo numérico, orientado a matrices y vectores. Por tanto desde el principio hay que pensar que todo lo que se pretenda hacer con él, será mucho más rápido y efectivo si se piensa en términos de matrices y vectores.

Según las especificaciones dadas, MATLAB 7.0.1 se eligió como entorno de desarrollo para la realización de este proyecto.



**Ilustración 10 : Interfaz de MATLAB 7.0.1 y toolbox GUIDE**

La utilización de MATLAB en el desarrollo de la aplicación impone ciertas restricciones sobre los recursos gráficos disponibles (tipos de botones, barras de desplazamiento, menús desplegables, etc.), restricciones que en ningún caso afectan a la funcionalidad requerida.

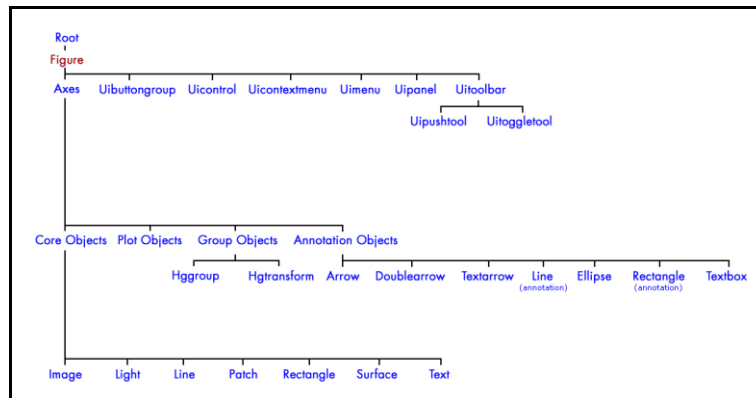
Para desarrollar la aplicación se necesitaban las siguientes herramientas :

- Una interfaz gráfica: desarrollado a partir de la *toolbox GUIDE* (Ilustración 10)
- Establecer un protocolo de comunicación entre el usuario y las cámaras a través de la interfaz del apartado anterior , es decir darle funcionalidad a la interfaz. Para desarrollar estos protocolos se utiliza el *software* de desarrollo aportado por la empresa productora de las cámaras : MotionPro X Camera SDK (Software Development Kit) .Ademas se necesita utilizar MotionPro X Timing Hub SDK para configurar el *Timing Hub* desde la interfaz.
- Un paquete de funciones para tratamiento de imágenes: se ha hecho uso del *Image Processing Toolbox*.

### 3.2.1.1.GUIDE (GRAPHICAL USER INTERFAZ DESIGN ENVIRONMENT)

Herramienta de desarrollo que permite editar interactivamente archivos *.fig* . Estas herramientas simplifican el proceso de diseño y construcción de GUIs. Se puede usar las herramientas de GUIDE para diseñar el *Layout* , o disposición del interfaz. Utilizando el editor de GUIDE se puede crear fácilmente la interfaz pulsando y

arrastrando componentes GUI (paneles , botones, campos de texto, *sliders* , menús ... ver Ilustración 11) . GUIDE guarda la interfaz en un archivo con extensión *.fig* .



**Ilustración 11 : Jerarquía de objetos utilizados por MATLAB para la implementación de interfaces**

Después de haber diseñado el *layout* y establecidas las propiedades de los componentes (color posición, estado...), el siguiente paso es programar la interfaz, implementando uno o mas *callbacks* en respuesta a eventos asociados a cada componente objeto. Los *callbacks* son funciones que se ejecutan como respuesta a alguna acción del usuario sobre un objeto. Como ejemplo típico es la acción de pulsar un botón.

Los *callback* se encuentran en un archivo *.m* que GUIDE genera automáticamente, este archivo esta relacionado con la interfaz con el mismo nombre pero con extensión *.fig* . GUIDE añade plantillas para las *callbacks* mas utilizadas, pero se pueden añadir otras. Se necesita el editor de MATLAB para modificar estos archivos.

### 3.2.1.2.SDK DE LAS CÁMARAS

El SDK de las cámaras incluye una interfaz de programación (API) que permite adquirir imágenes y controlar las cámaras desde una aplicación desarrollada con MATLAB. La interfaz trabaja con MATLAB 6.5 o superior y no soporta Windows NT.

La interfaz incluye las funciones que manejan las camaras comprimidas en una librería llamada *XStreamML.dll* junto con un archivo específico para Matlab denominada *XStreamML.m*. Todas las rutinas pueden ser invocadas desde un *script* de MATLAB utilizando la siguiente sintaxis:

**[output1, output2 ...] = XStreamML [input1, input2 ...]**

El número de entradas (*inputs*) y salidas (*outputs*) depende de la función utilizada. Mas detalles de la sintaxis de los comandos puede ser consultada escribiendo “help XStreamML” en la línea de comando o bien abriendo el archivo *XStreamML.m* con un editor de texto.

A continuación se explica cómo usar, a grandes rasgos, la interfaz de programación:

Antes de llamar cualquier función las cámaras deben estar *abiertas*. Para *abrir* una cámaras, el usuario invoca a la función

**[nResult, nCameraId] = XStreamML ('OpenCamera', nInputId)**

donde '*nInputId*' es el ID de la cámara o 0 para abrir la primera cámara disponible. Para obtener una lista de todas las cámaras disponibles, se llama a la función

**[nResult, nItems, svArray] = XStreamML ('EnumCameras')**

Para configurar cada parámetro de cada cámara , se realizan una llamada a la función

**[nResult] = XStreamML ('SetParameter', nCameraId, nParamKey, nValue)**

Donde se indica con *nParamKey* , el parámetro , *nValue* , el valor y *nCameraId* , el ID de la cámara. . Cuando han sido establecidos los parámetros se invoca el método:

**[nResult] = XStreamML ('SendCfg', nCameraId)**

para enviar a las cámaras la nueva configuración y activarla. Si se desea conocer el valor de algún parámetro se puede llama a la función:

**[nResult, nValue, nMinValue, nMaxValue] = XStreamML ('GetParameter', nCameraId, ParamKey)**

Una vez abiertas las cámaras, con la siguiente función se adquiere una imagen. Al invocar esta función , la cámara permite la entrada de luz hacia el sensor un tiempo *nTimeOut*. *Se genera una imagen que se devuelve como resultado. La segunda imagen devuelta se utiliza para un modo de grabación especial, en esta aplicación no se ha implementado tal modo.*

**[nResult,image1,image2] = XStreamML ('SynchGrab', nCameraId, nTimeOut)**

Para realizar una captura de un numero determinado de imágenes según los parámetros configurados previamente en la cámaras , se debe llamar un método que indique el comienzo de la grabación dentro de la memoria de la cámara y el numero de imágenes a capturar entre otros parámetros. Esta funcion indica a la cámara que comience la grabacion según los parámetros establecidos.

**[nResult] = XStreamML ('MemoryStartGrab', nCameraId, nStartAddLo, nStartAddHi, Frames, nPreTrigFrames)**

Cuando se invoca el siguiente método , se conoce el estado actual de la grabación en la cámara seleccionada . Si *nResult* tiene un valor de 1 , indica que la cámara ha finalizado la grabación, un valor de 0 indica que la cámara o bien continua capturando y grabando imágenes a la memoria de la cámara o bien está aún en espera del *trigger*.

**[nResult] = XStreamML ('GrabsReady', nCameraId)**

Se puede cancelar la adquisición en curso y leer imágenes grabadas en la memoria de la cámara con las rutinas:

**[nResult] = XStreamML ('MemoryStopGrab', nCameraId)**

```
[nResult, image] = XStreamML ('MemoryReadData', nCameraId,
nlsFirst, nStartAddLo, startAddHi, nFrameIdx)
```

En el documento La interfaz de MATLAB , para todas las rutina , devuelve un código de error que se puede consultar en el documento [1]

### 3.2.1.3.IMAGE PROCESSING TOOLBOX

El *Image Processing Toolbox* es una colección de funciones que extienden la capacidad de computo numérico de MATLAB. Esta herramienta aporta un gran número de operaciones para el proceso de imágenes, incluyendo filtrado espacial, transformaciones morfológicas, operaciones locales, operaciones de bloque, diseño de filtros, mejora de imágenes, etc.

Las imágenes en Matlab son representadas como *arrays* de 1 o 3 dimensiones, según ésta sea una imagen en escala de grises o a color. Cada valor puede ser representado en formato entero (por ejemplo *uint8*, valores posibles entre 0 y 255) o decimal (*Double*, valores posibles entre 0 y 1).

Este *Toolbox* incluye funciones para el manejo de archivos de imagen como lectura o escritura, además de una colección de métodos para realizar operaciones entre imágenes y transformación de RGB a escala de grises, lectura y creación de archivos de imágenes.

También es posible aplicar operadores morfológicos, operadores de bloque y locales, diseñar y aplicar filtros lineales específicos para imágenes, calcular la *Trasformada de Fourier* y la *DCT* para su posterior filtrado en frecuencia, mejora para el análisis de textura (por ejemplo) de imágenes, registrar imágenes para la composición de panorámicas, realizar transformaciones espaciales como interpolación , escalado y recorte de imágenes, etc.

Este *Toolbox* ha sido utilizado principalmente en la etapa de desarrollo de las herramientas destinadas a la mejora de la secuencia resultante, explicadas en detalle en capítulo 4.

## 3.3. Descripción de la aplicación

### 3.3.1. Introducción

En este apartado se describe la funcionalidad de la aplicación desarrollada, teniendo en cuenta las especificaciones iniciales además de las limitaciones del sistema. Para una descripción mas detallada de cada uno de los objetos que componen cada interfaz consultar el manual de usuario, ANEXO A .

Parece razonable idear una única solución para resolver ambos problemas ya que los dos comparten muchísimas similitudes. Pero para una mayor organización y simplicidad, se prefirió solucionar de manera individual los dos problemas planteados

de modo que el resultado son dos sub-aplicaciones separadas, con bastantes partes comunes pero con pequeñas diferencias: disposición de los objetos en el marco, configuración del modo de grabación, algoritmo de alineamiento...

A su vez en cada una de estas sub-aplicaciones o *modos* de captura se definen dos etapas para la obtención de la secuencia deseada:

- Configuraron y alineamiento de las cámaras.
- Captura y descarga de la secuencia a disco.
- Además se añade una aplicación para el análisis cinemático de la secuencia resultante, que a diferencia de los dos puntos previos, sería común a ambos modos ya que se puede utilizar con cualquier secuencia de imágenes,

Al tratarse de una aplicación con fines comerciales y desconocer el tipo de usuario que lo va a utilizar, se pone especial énfasis en desarrollar una aplicación sencilla de manejar, con un entorno *amigable* y la completa funcionalidad que permite el SDK [1] de la cámaras utilizadas.

Debido a la similitud entre ambas aplicaciones, a continuación se describe las principales funciones de la interfaz implementada para el modo *AumentoTasa* y posteriormente se exponen los cambios introducidos para desarrollar el modo *Panorámic*. Por ultimo se describen las características de la aplicación destinada al estudio físico de secuencia, denominado *Cinematic Analysis*

#### **3.3.2. Modo *AumentoTasa*: Asistente de alineamiento y ajuste**

Esta aplicación ha sido desarrollada para configurar los parámetros del sistema de cámaras tales como tiempo de exposición, *binning*, ganancia del sensor, ROI ... Además se utiliza para indicar el orden que se utilizará para intercalar las imágenes de cada una de las cámaras y componer la secuencia final. También es posible ejecutar un algoritmo desarrollado para calcular la región de interés común a todas las cámaras implicadas en el proceso de captura, consultar capítulo 4.4.2.

La siguiente figura , Ilustración 12 , muestra una visión general de la interfaz,. En cada marco se puede visualizar las imágenes capturadas por cada cámara. Los cuadros de control permiten configurar los parámetros, seleccionar la ROI, visualizar las cámaras en directo.



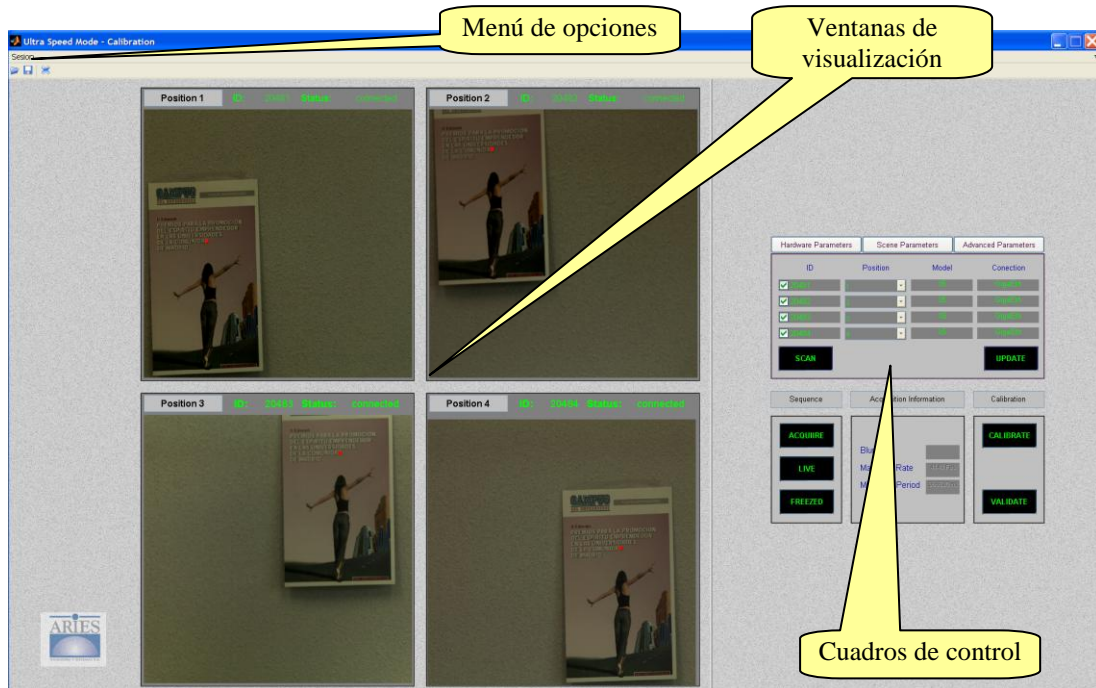


Ilustración 12 : Aspecto general del interfaz de ajuste y alineamiento para el modo *AumentoTasa*

### 3.3.3. Modo *AumentoTasa*: Herramienta de captura y montaje

Esta herramienta (ver Ilustración 13) permite configurar las variables que determinan las características de la secuencia final. Además una vez realizada la captura es posible descargar todas o una parte de las imágenes de la secuencia final a disco. En una futura mejora, esta herramienta incluirá también una interfaz para la generación de una secuencia mejorada utilizando las herramientas implementadas según el capítulo 4.

Una vez realizado el alineamiento y ajuste de parámetros, y antes de proceder a la grabación, es interesante poder previsualizar el aspecto que tendrá la secuencia resultante, sin tratamiento alguno, para poder confirmar que la etapa anterior está bien realizada. Con esta visualización se detectan las diferencias de enfoque, apertura del obturador y punto de vista entre las cámaras. Además esta herramienta ofrece la posibilidad de realizar un ajuste mas fino sobre las lentes de las cámaras (apertura del obturador y enfoque) para que las características de los sistemas ópticos utilizados sean lo mas parecidas posible. Después de este “ajuste” fino se puede proceder a la captura, pre-visualización y descarga a disco de parte o la totalidad de las imágenes de la secuencia capturada.

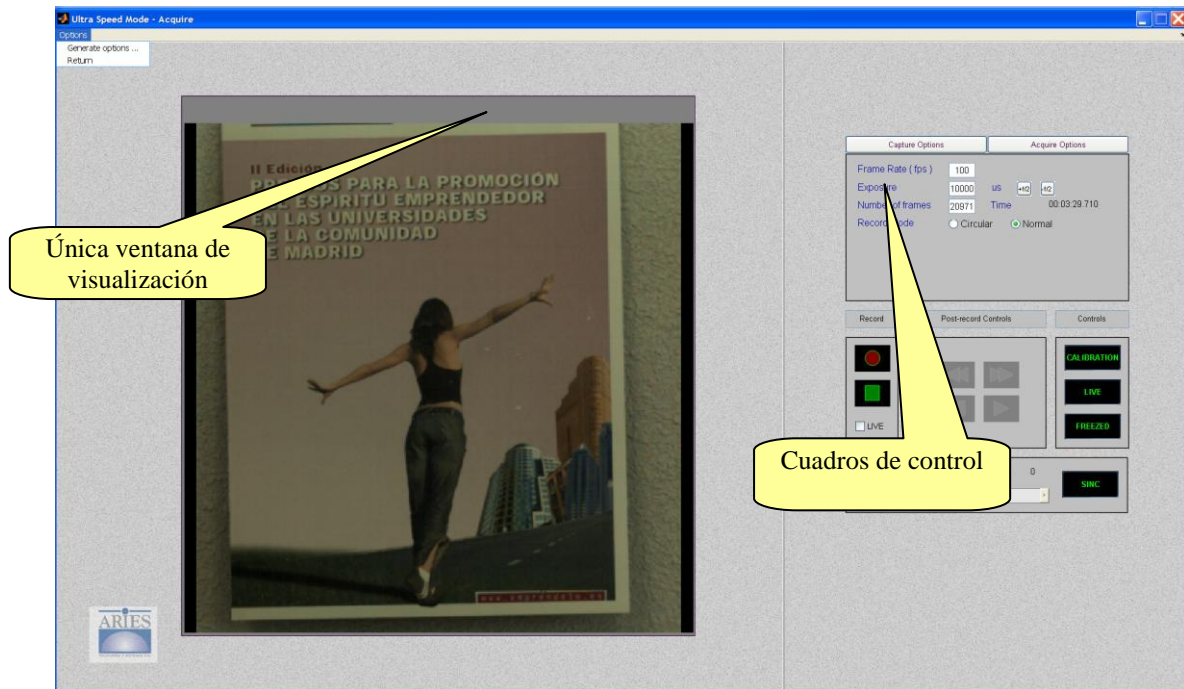
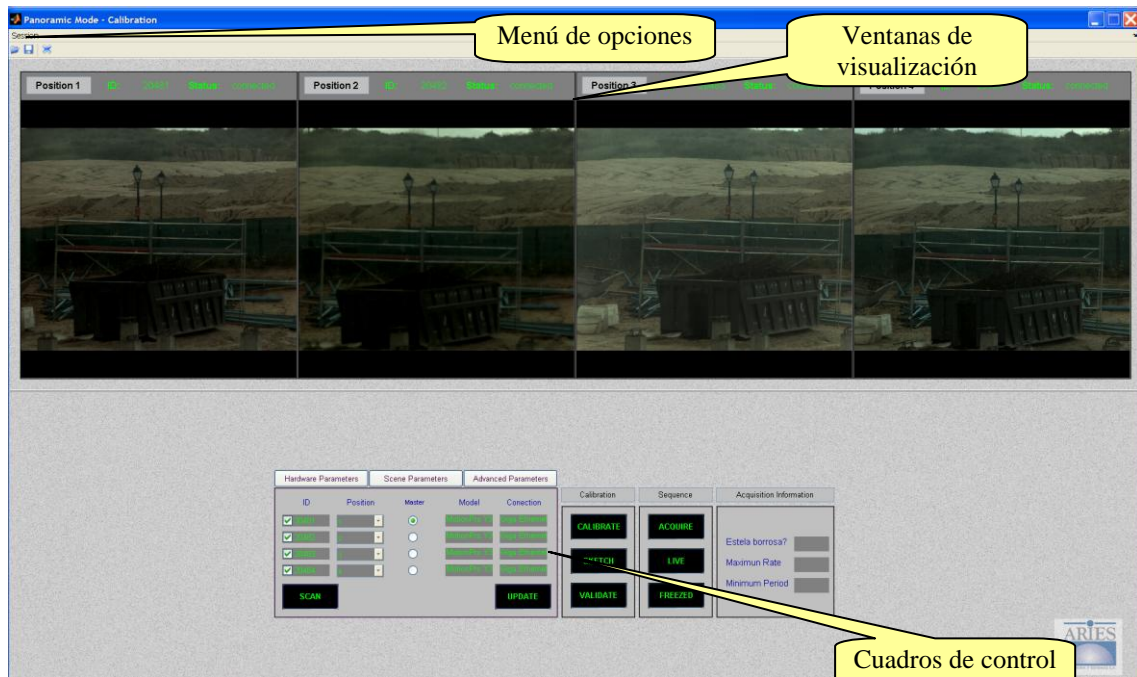


Ilustración 13 : Aspecto general del interfaz de captura y montaje

### 3.3.4. Modo *Panorámico*: Asistente de alineamiento y ajuste

Esta aplicación ha sido desarrollada para configurar los parámetros del sistema de cámaras en el modo *AumentoCampo*. Hay que tener en cuenta que la mayoría de los controles desarrollados en esta aplicación son idénticos a los desarrollados para el modo *AumentoTasa*. Por esto la funcionalidad de la aplicación es la misma salvo algunos detalles que se comentan en el ANEXO A.

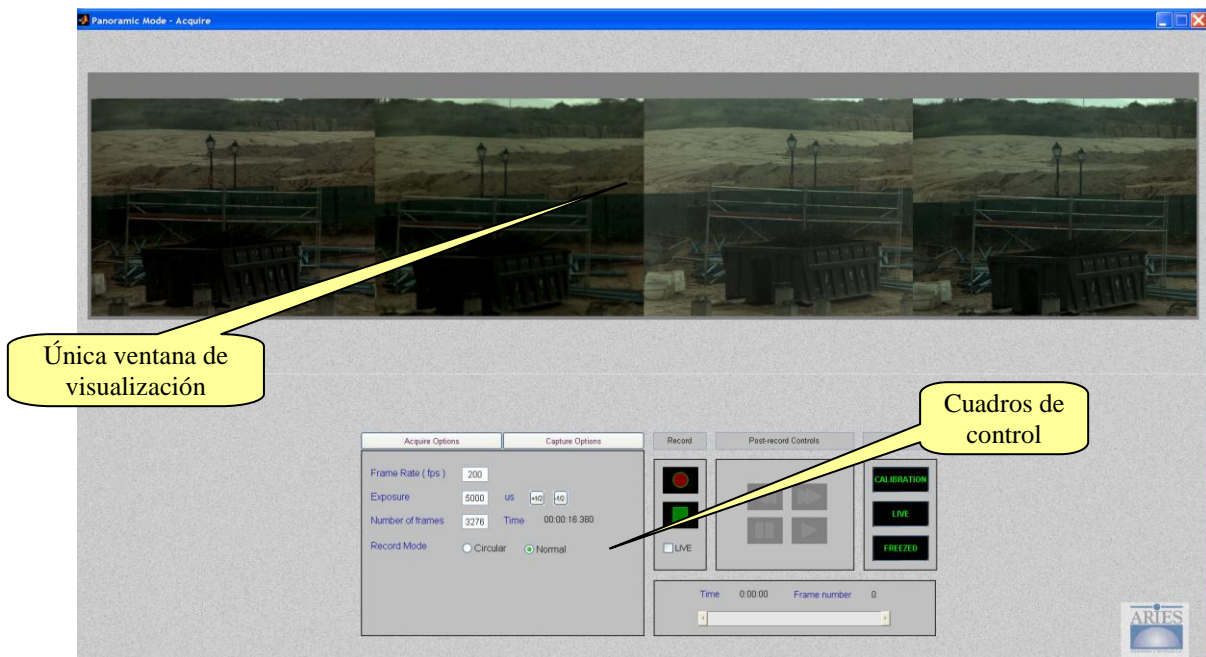
Si se piensa en el sistema, ambos modos difieren en el sincronismo de la señal de captura. El resto de componentes (*trigger*, cámaras, conexiones...) es idéntico. Otra diferencia es la ubicación de las cámaras con respecto a la escena a capturar (no incumbe al *software* utilizado) y el asistente de alineamiento desarrollado para el modo *AumentoTasa*.



**Ilustración 14 :** Aspecto general de la interfaz de alineamiento y ajuste para el modo *AumentoCampo*.

### 3.3.5. Modo AumentoCampo: Herramienta de captura y montaje

Esta herramienta, al igual que la desarrollada para el modo *AumentoTasa*, permite configurar las cámaras para realizar una captura. Los parámetros configurables en esta aplicación están relacionados con la secuencia final deseada como *fps*, *Exposure*, número de imágenes, modo de grabación... además de poder realizar una grabación, se puede visualizar parte de la secuencia capturada y descargarla a disco.



**Ilustración 15 : AumentoCampo , interfaz de captura y montaje . Pan\_adquisicion.fig.**

Aparte de las modificaciones realizadas en la colocación de los objetos, que se muestran en la Ilustración 15, en esta fase no se introdujeron nuevos objetos al diseño creado para el modo *AumentoTasa*.

Las modificaciones hechas en el código asociado para adaptarla al modo de grabación deseado se explicarán mas adelante en el apartado 3.4 , funcionamiento de la aplicación.

### 3.3.6. Cinematic Analysis

Tal y como se expone en los requisitos iniciales, se ha desarrollado una herramienta para obtener estimaciones de los parámetros de movimiento del objeto filmado. Para seleccionar el objeto, el usuario selecciona marcando puntos móviles, puntos de giro y líneas de referencia sobre los objetos de la imagen. Esta aplicación es independiente de las anteriores y se utiliza *off-line* ya que el sistema de cámaras en esta etapa no es necesario.

Como punto de partida, esta aplicación necesita una secuencia de imágenes, numeradas, con un nombre común, y ubicadas en la misma carpeta. Esta secuencia es el resultado de las etapas de captura, montaje y descarga, para uno de los dos modos.

Para poder obtener mediciones aproximadas de la escena (obsérvese que se está trabajando con cámaras no calibradas), es necesario que un usuario marque referencias sobre algunas de las imágenes de la secuencia (mínimo 1 punto en al menos 3 imágenes). Además es necesario conocer el numero de cuadros por segundo (fps) de la secuencia analizada y una relación píxel/longitud para poder ofrecer las medidas en



unidades de la escena. La Ilustración 16 muestra el interfaz desarrollado. A continuación , se describe la interfaz y su funcionalidad



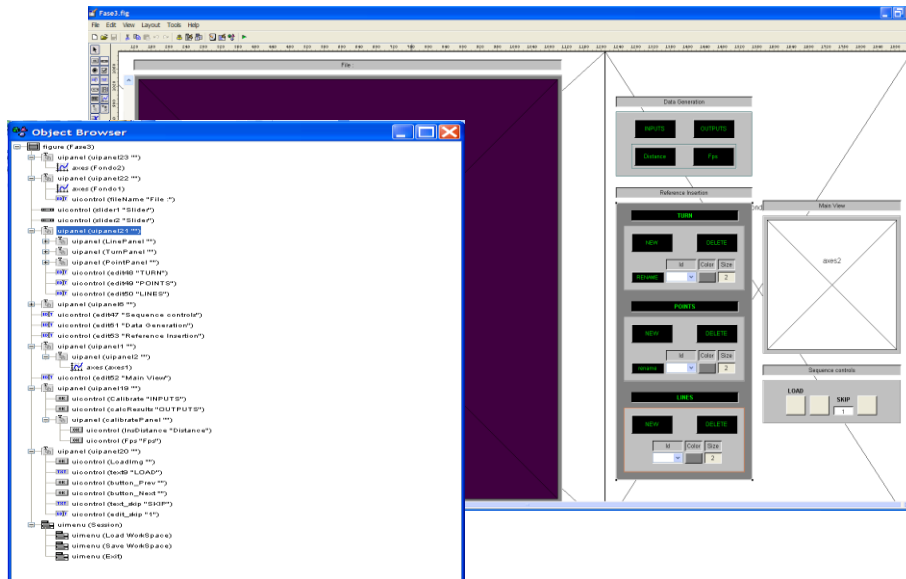
Ilustración 16 : Aspecto general de la interfaz de *Cinematic Analysis*

### 3.4. Funcionamiento de la aplicación

#### 3.4.1. Introducción

En este apartado se documenta y desarrolla, mediante esquemas y tablas, los procedimientos utilizados para implementar cada una de las aplicaciones, de modo que sea inmediato localizar el código necesario para modificar o añadir una funcionalidad. Para cada modo, *AumentoTasa* y *AumentoCampo*, así como para la aplicación de *Cinematic Analysis*, se ha elaborado un listado de los archivos utilizados por cada apartado, así como una descripción de las funciones utilizadas y un diagrama de ejecución de la aplicación. Para profundizar más en alguno de los temas, consultar directamente el código implementado.

Éstas aplicaciones fueron realizadas utilizando el *toolbox* de MATLAB para la creación de interfaces graficas (GUIDE). Al crear una interfaz grafica con esta herramienta, se genera un archivo *.fig* donde se encuentra el diseño de la ventana con los objetos gráficos colocados en ella, es decir la interfaz gráfica. Este archivo va asociado a otro archivo con extensión *.m* y con el mismo nombre, donde se implementa el código y funciones *callback* de los objetos del interfaz. Estos objetos se organizan en una estructura en árbol donde la raíz es denominada *handles*.



**Ilustración 17 : Interfaz grafica desarrollada con GUIDE y jerarquía de objetos**

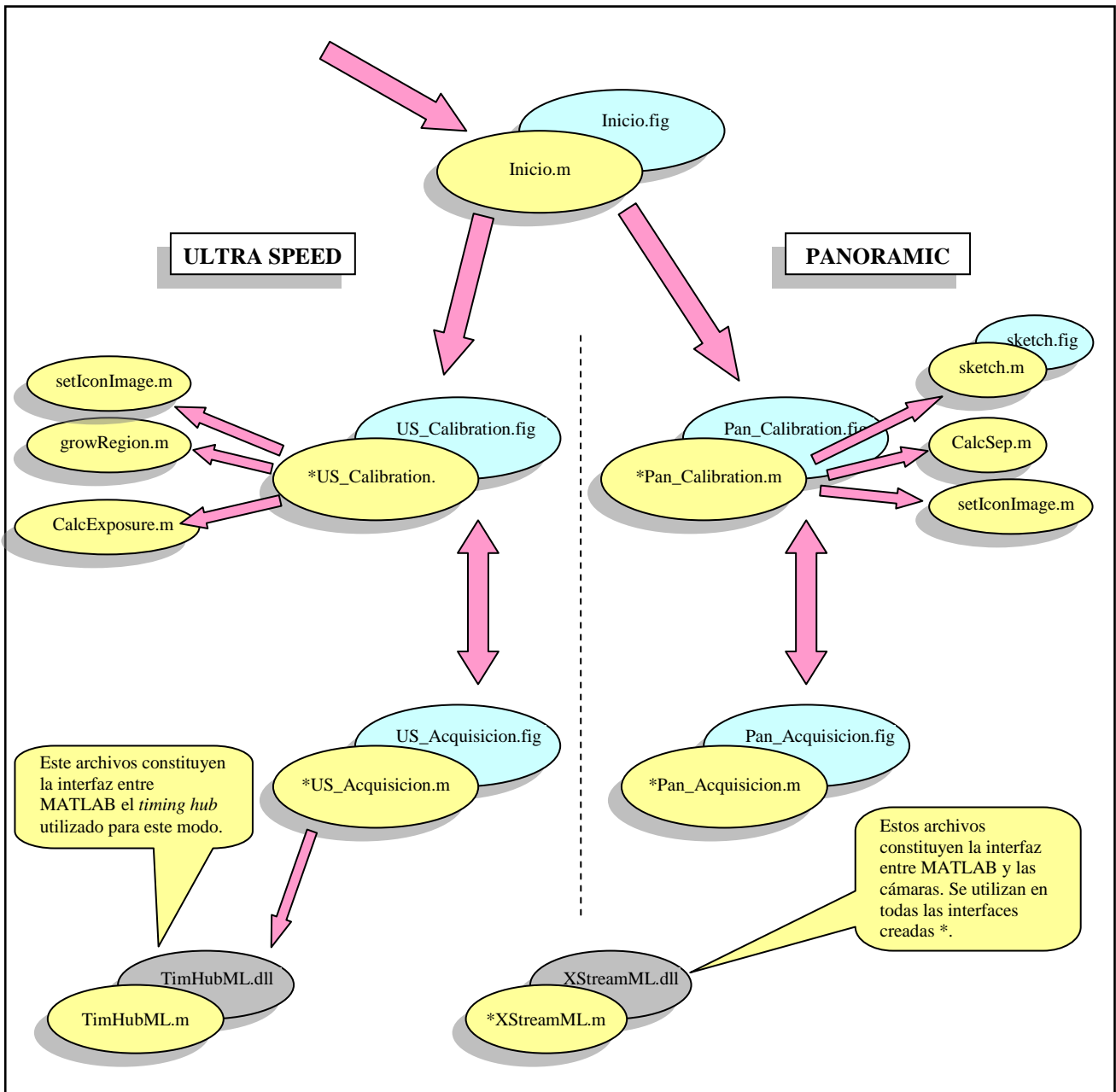
Dentro de la estructura de *handles* también se pueden introducir variables útiles para el funcionamiento de la aplicación y utilizadas por el programador. Para que estas variables se queden grabadas en la estructura y/o las modificaciones de las propiedades de los objetos sea efectiva, después de haberlas realizado, tiene que ejecutarse la instrucción

```
guidata(hObject, handles);
```

### 3.4.2. Flujo de las aplicaciones

Las siguientes imagenes, Ilustración 18 e Ilustración 19 , representan diagramas del flujo de las aplicaciones. El desarrollo de este Proyecto ha sido separado en varias aplicaciones, se presenta un diagrama para cada modo de operación: modo *AumentoTasa*, modo *AumentoCampo* y *Cinematic Analysis*.

En estos diagramas existe un flujo que no ha sido representado específicamente mediante flechas, y que sin embargo resulta ser el más frecuente. Se trata del flujo producido entre las interfaces (archivos *.fig* ) y el código ligado a ellas (archivos *.m* ) . Esta relación se produce cuando el usuario genera acciones y éstas invocan funciones desarrolladas dentro del código legible y editable. Para representar estas relaciones en la Ilustración 18 han sido colocados de manera próxima estos archivos, utilizando un color turquesa para representar las interfaces y tonos ocre para representar los *scripts*.



**Ilustración 18 : Flujo de ejecución para las aplicaciones en modo *AumentoTasa* y *AumentoCampo*.**

En la aplicación de *Cinematic Analysis*, casi todo el flujo de la aplicación, representado en la Ilustración 19, se produce dentro del archivo *Fase3.m* y *Fase3.fig* . La iteración entre del usuario y la interfaz produce la ejecución de funciones, denominados *callback* , en el archivo con extensión *.m*.

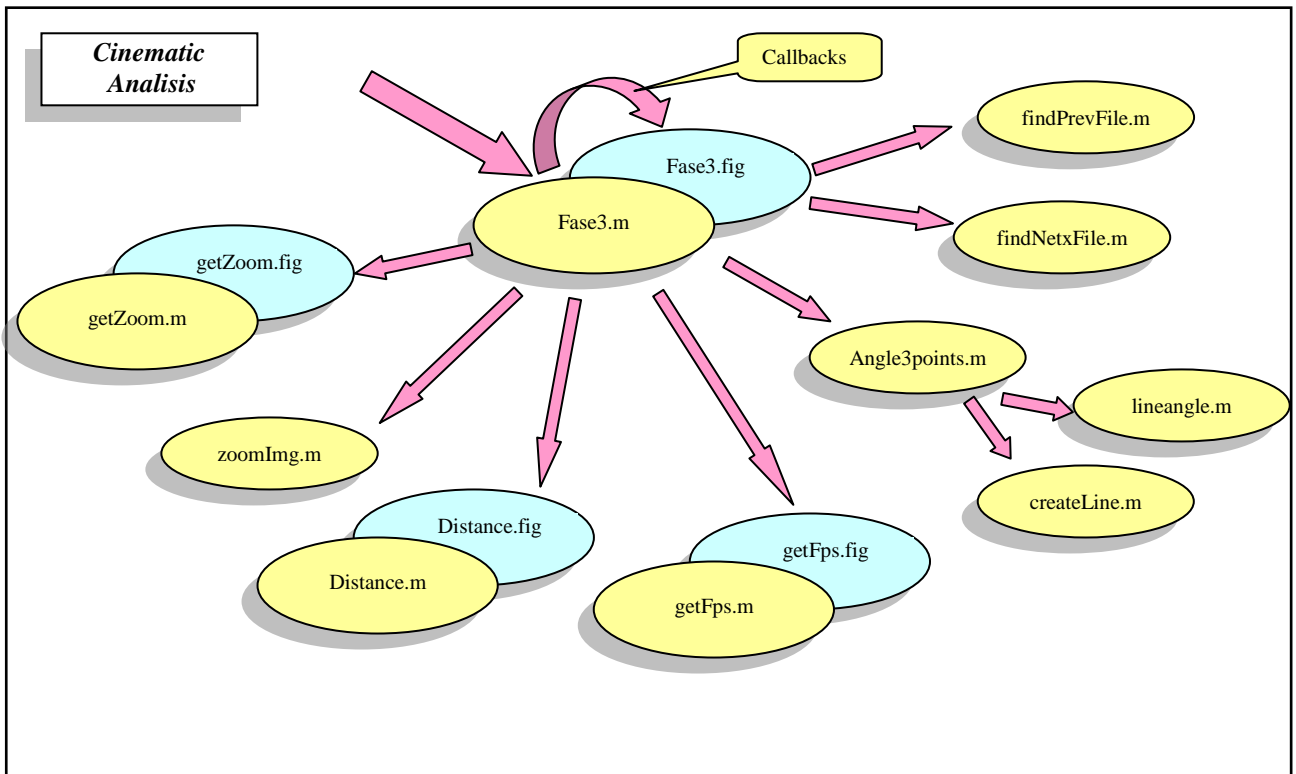


Ilustración 19 : Flujo de ejecución de la aplicación *Analisis Cinematic*

### 3.4.3. Variables y estructuras utilizadas

En este apartado se describen las variables y estructuras más importantes utilizadas para poder ejecutar y desarrollar la funcionalidad de las aplicaciones. Estas variables y estructuras deben permitir el acceso y actualización desde cualquier función de la interfaz incluso desde fuera de ella; por este motivo son representadas como variables globales (*global nombreDeLaVariable*) o como un objeto dentro de la jerarquía de la interfaz (*handles.nombreDeLaVariable*).

#### 3.4.3.1.MODO AUMENTOTASA Y AUMENTOCAMPO

El conjunto de las aplicaciones que permiten ejecutar ambos modos poseen estructuras de datos y variables globales idénticas así que las describiremos conjuntamente.

Por un lado tenemos variables globales que describen las características del sistema:

- *g\_NumCam*: numero de cámaras detectadas por la aplicación.
- *g\_idCam*: identificador de las cámaras.
- *g\_Pos*: posición en la estructura.
- *g\_Type* : cámara master (1) o esclava(0).Esta variable tan solo se utiliza para la sincronización en el modo *AumentoCampo*



- *g\_CameraOn*: indica si la cámara esta activa para el modo *live*.

También se utilizan variables globales para controlar la reproducción de la secuencia capturada o visualizar el modo *Live* :

- *stopRec* : indica que la grabación ha sido cancelada por el usuario.
- *g\_NumImgCapture* : indica el número de imágenes capturadas de la secuencia.
- *idx* : indica el frame actual que se esta visualizando.
- *g\_nFrameIdxTrig* : índice del *frame* donde se produjo el *trigger*.
- *g\_nFramePreTrig*: numero de imágenes que capturar antes de la señal de *trigger*.
- *g\_IdHub* : identificador del *timing hub*, *unicamente se utiliza para la sincronización del modo AumentoTasa*.
- *g\_play* : variable utilizada para el modo *live*.
- *g\_pixSol* : numero de píxeles de solape (sólo se utiliza en el modo *Panorámic*).

Por otro lado se crearon estructuras para almacenar la configuración de las cámaras y las características de la *escene*, para facilitar el manejo de estos parámetros (en las opciones de carga y guardar de los parámetros configurados). Concretamente se idearon dos estructuras:

- En la primera estructura se almacena la configuración de cada cámara; se crea una estructura de este tipo para cada cámara conectada. A continuación se muestra los distintos campos de la estructura , el parámetro *i* indica el numero de cámara

```
param(i)=struct('Cam_Status', ' ', ...
               'Cam_Pos', i, ...
               'Exposure', 5000000, ...
               'Period', 5005000, ...
               'GainCam', 0, ...
               'GainPix', 0 , ...
               'Binning', 0);
```

- La segunda estructura almacena los parámetro relativos a la escena a grabar:

```
scene=struct('Distance', 50, ...
            'UDistance', 1, ...
            'Lens', 1, ...
            'Speed', 100, ...
            'USpeed', 1, ...
            'Fps', 200);
```

A estas estructuras se accede a través del objeto gráfico *padre* (*handles*).

### 3.4.3.2.APLICACIÓN DE *CINEMATIC ANALYSIS*

En esta aplicación se utilizan las siguientes variables globales:

- *g\_numPoints*: indica el número máximo de puntos por imagen.
- *g\_numTurn*: indica el número máximo de puntos de giro.
- *g\_numLines*: indica el número máximo de líneas de referencia.

Además se utilizan un tipo de estructura denominada *cell* para almacenar las coordenadas y características de los objetos utilizados:

- *ArrayP* : Almacena las características del objeto , puntos , y sus coordenadas en cada una de la imágenes.
- *ArrayL* : Almacena las características del objeto , línea , y sus coordenadas.
- *ArrayT* : Almacena las características del objeto , punto de giro , y sus coordenadas.

Para poder acceder a estas estructuras desde cualquier parte de la aplicación , se almacenan dentro de las variables asociadas a la interfaz, concretamente *cuelgan* del objeto *padre* denominado *handles*

### 3.4.4. AumentoTasa

En este apartado, se describen cada uno de los archivos utilizados para ejecutar este modo. Además para las interfaces principales se explican las funciones desarrolladas como respuesta a las acciones ejecutadas sobre los objetos de la interfaz, para la configuración de las cámaras y para organizar la interfaz.

#### 3.4.4.1.DESCRIPCIÓN DEL SISTEMA DE ARCHIVOS

La Tabla 2 muestra un listado de los archivos implementados para desarrollar este modo. Incluyen ambas etapas: asistente de alineamiento y ajuste y la herramienta de captura y montaje. Las referencias mencionadas en esta tabla están relacionadas con la Ilustración 20.

Tabla 2: Sistema de archivos para el modo *AumentoTasa*

Nombre	Comentarios
US_Calibration.fig	Interfaz Grafica asociado a <b>US_Calibration.m</b> , fase de alineamiento y ajuste de parámetros .
US_Calibration.m	
US_Acquisicion.fig	Interfaz Grafica asociado a <b>US_Acquisition.m</b> , fase de captura y montaje.
US_Acquisicion.m	
TimHubML.dll	Extensión de la aplicación
TimHubML.m	
XStreamML.dll	Extensión de la aplicación
XStreamML.m	
Menu_File_Exit_Callback.m	Referencia 3 en la , funciones implementadas <b>CloseCameras</b> , comentada en <b>US_calibration</b> . Finaliza la aplicación

<b>Menu_File_Load_Callback.m</b>	Referencia 1, funciones utilizadas SendParameter2Cam y ShowImg , comentadas en US_calibration.m . Esta función carga los parámetros de la configuración de las cámaras
<b>Menu_File_Save_Callback.m</b>	Referencia 2, esta función guarda los parámetros de configuración de las cámaras
<b>setIconImage.m</b>	Función que crea el menú de herramientas
<b>CalcExposure.m</b>	Función que calcula el tiempo de exposición mínimo y la estela borrosa
<b>CalcROI.m</b>	Función que calcula de manera teórica la región común
<b>growRegion.m</b>	Algoritmo de crecimiento de regiones
<b>imágenes</b>	Directorio con las imágenes de fondos de la aplicación
<b>Icons</b>	Directorio con los iconos utilizados en la barra de herramientas y el reproductor

### 3.4.4.2.DIAGRAMA DEL ASISTENTE DE ALINEAMIENTO Y AJUSTE

En este apartado se describen las funciones implementadas en el archivo *US\_Calibration.m* que se corresponde con la etapa de alineamiento y ajuste para el modo *AumentoTasa*. Estas funciones pueden ser separadas en dos grandes grupos: *callbacks* , invocadas desde la interfaz , y funciones auxiliares , implementadas para manejar las cámaras , implementar algoritmos , realizar operación auxiliares ... La siguiente imagen , Ilustración 20 , muestra algunos detalles de la interfaz . Además se numeran los objetos que tienen asociada una función descrita en la Tabla 3.

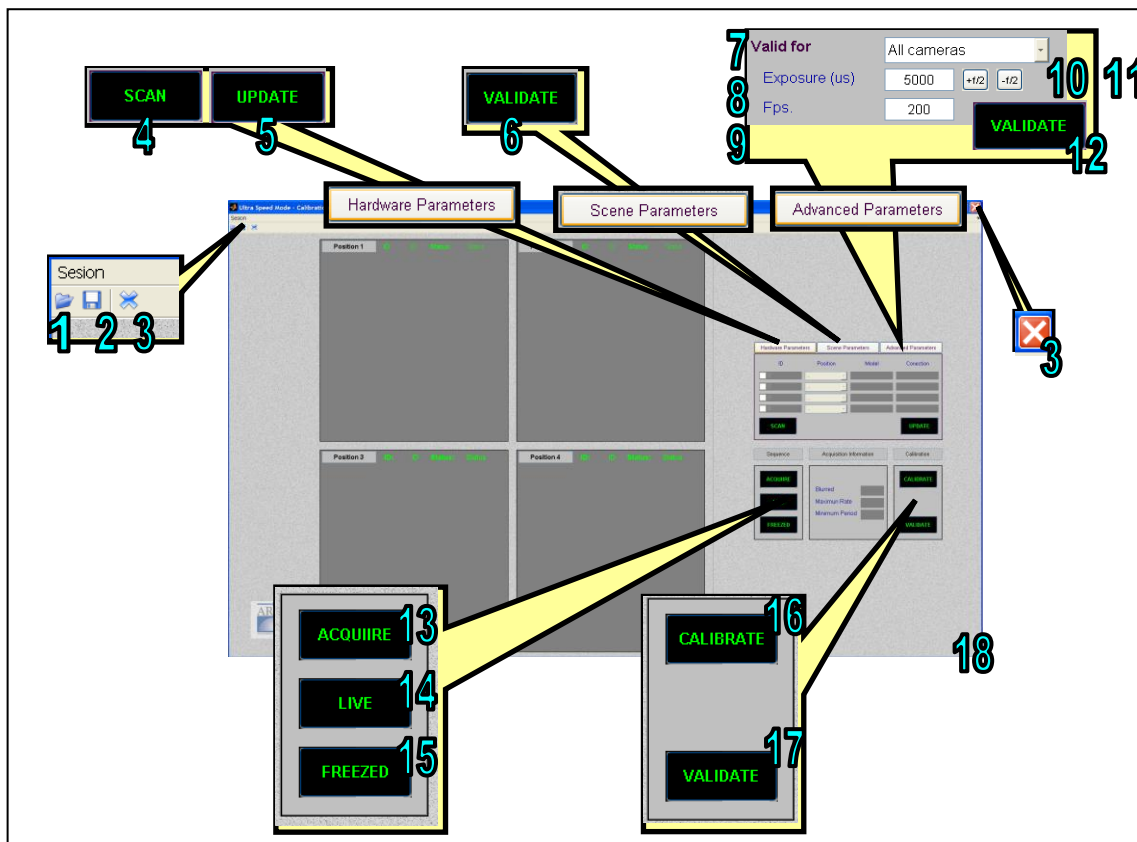


Ilustración 20 : Interfaz de Alineamiento y ajuste del modo *AumentoTasa* : *US\_Calibration.fig*

Tabla 3 : Métodos implementados en el archivo US\_Calibration.m

Ref	Funciones		Funciones invocadas	Descripción
	Nombre	Comentarios		
	<b>US_Calibration</b>			
19	<b>US_Calibration_OpeningFcn</b>	Función que se ejecuta al invocar la función <b>US_Calibration</b> antes de que se haga visible	<b>InitStruct</b> , <b>ShowImg</b> , <b>SenParameters2</b> , <b>Cam</b> , <b>OpenCameras</b> , <b>setIconImage.m</b> , <b>XStreamML.m</b>	Inicia la aplicación , se ejecutan rutinas para iniciar variables globales
	<b>US_Calibration_OutputFcn</b>			
4	<b>Scan_Callback</b>	Callback del botón SCAN ( <i>Hardware Parameters</i> )	<b>SendParameter2</b> , <b>Cam</b> , <b>ShowImg</b> , <b>OpenCameras</b> , <b>InitStruct</b> , <b>XStreamML.m</b>	Refresca las conexiones
5	<b>Update_Callback</b>	Callback del botón UPDATE ( <i>Hardware parameters</i> )	<b>ShowImg</b>	Muestra imágenes de cada cámara
6	<b>UpdateScene_Callback</b>	Callback del botón VALIDATE ( <i>Scene Parameters</i> )	<b>ShowImg</b> , <b>CalcROI.m</b> , <b>XStreamML.m</b>	Actualiza los parámetros de escena
7	<b>CameraldAdvanced_Callback</b>	Callback del desplegable VALID FOR ( <i>Advanced parameters</i> )		Muestra los parámetros configurados para cada cámara, además sirve para modificar los parámetros avanzados de todas o alguna cámara.
12	<b>UpdateAdvanced_Callback</b>	Callback del botón VALIDATE ( <i>Advanced parameters</i> )	<b>SendParameter2</b> , <b>Cam</b> , <b>ShowImg</b> , <b>CalcExposure.m</b> , <b>XStreamML.m</b>	Actualiza los parámetros configurados en este panel para la/s cámara/s indicadas en VALID FOR
10	<b>Ma_Exposure_Callback</b>	Callback del botón + f / 2 ( <i>Advanced parameters</i> )	<b>Exposure_Callback</b>	Aumenta la luminosidad ( <i>exposure</i> ) en un paso
11	<b>Me_Exposure_Callback</b>	Callback del botón - f / 2 ( <i>Advanced parameters</i> )	<b>Exposure_Callback</b>	Disminuye la luminosidad ( <i>exposure</i> ) en un paso
8	<b>Exposure_Callback</b>	Callback del edit EXPOSURE ( <i>Advanced parameters</i> )	<b>XStreamML.m</b>	Controla que el tiempo de Exposición no se establezca fuera de los límites además que actualiza el parámetro en las cámaras
9	<b>Fps_Callback</b>	Callback del edit Fps ( <i>Advanced parameters</i> )	<b>XStreamML.m</b>	Controla que el número de fps no se establezca fuera de los límites y modifica el valor de exposure ya que están relacionados
14	<b>Play_Callback</b>	Callback del botón LIVE ( <i>Sequence</i> )	<b>ShowImg</b>	Activa el modo LIVE
13	<b>Acquire_Callback</b>	Callback del botón ACQUIRE ( <i>Sequence</i> )	<b>US_Acquisition.m</b>	Invoca a la pantalla de captura y montaje de la secuencia , se pasan variables a través de <i>varargin</i>
15	<b>Stop_Callback</b>	Callback del botón FREZZED ( <i>Sequence</i> )		Finaliza el modo LIVE
	<b>InitStruct</b>			Inicializa parámetros de <i>exposure</i> , cámara activa , <i>ID</i> , <i>binning</i> , <i>sensor gain</i> , <i>pixel gain</i> para cada cámara conectada además de los parámetros de la escena : distancia , tipo de lente y

				velocidad aproximada y los almacena en una estructura
	<b>ShowImg</b>	<i>timeout</i> indica el tiempo máximo de espera de la cámara para capturar la imagen (ns), si se supera este tiempo se devuelve con <i>nResult</i> un valor distinto de 0	<b>XStreamML.m</b>	Muestra imágenes de las cámaras conectadas y activadas , utilizando la interfaz <b>XStreamML.m</b>
	<b>OpenCameras</b>		<b>XStreamML.m</b>	Detecta cámaras conectadas y crea variables para controlar las cámaras
	<b>SendParameter2Cam</b>		<b>XStreamML.m</b>	Actualiza los parámetros de las cámaras
16	<b>Cal_Auto_Callback</b>	Callback del botón CALIBRATE ( <i>Calibration</i> ) , Se utiliza <b>XStreamML.m</b> para obtener unos valores de las cámaras	<b>calcOriginDimension, ShowImg, XStreamML.m, growRegion.m</b>	Solicita al usuario que marque puntos para realizar el alineamiento
17	<b>Cal_End_Callback</b>	Callback del botón VALIDATE ( <i>Calibration</i> )		Implementa el algoritmo de alineamiento
	<b>calcOriginDimension</b>	Función auxiliar para el alineamiento	<b>distanceMin</b>	Esta función extrae las coordenadas de los puntos 'marcados' y calcula la ROI
	<b>distanceMin</b>	Función auxiliar para el alineamiento		Introduciendo las coordenadas de los puntos homólogos y tamaño de la imagen , calcula el origen para cada imagen y dimensión de la región común a las 4 imágenes
	<b>CloseCameras</b>	Función implementada en <b>Menu_File_Exit_Callback.m</b>	<b>XStreamML.m</b>	Invoca al método <b>XStreamML('CloseCamera',ID)</b> que cierra las cámaras previamente abiertas.

### 3.4.4.3.DIAGRAMA DE LA HERRAMIENTA DE CAPTURA Y MONTAJE

Al igual que en el aparatado anterior, utilizando una tabla se describen las funciones implementados de la interfaz principal de esta aplicación. Además se acompaña de la Ilustración 21 para identificar los objetos que producen la ejecución de cada función.

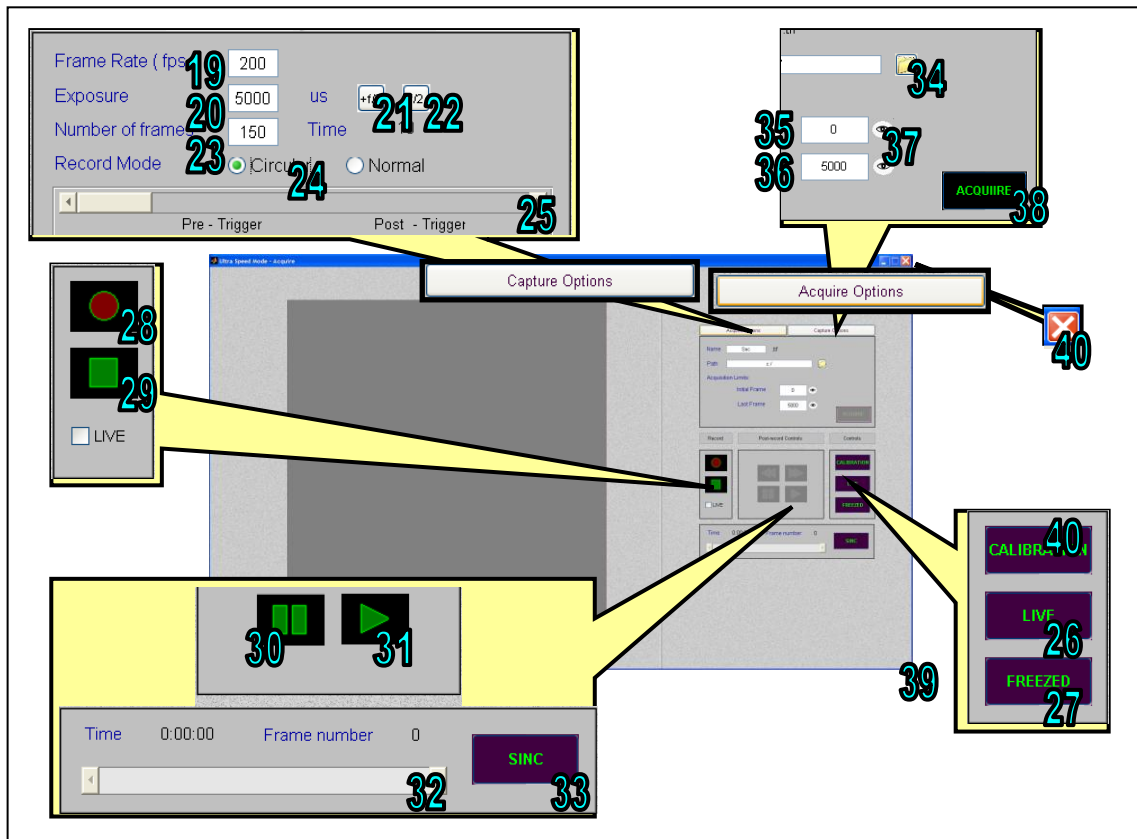


Ilustración 21 : Interfaz de Captura y montaje del modo *AumentoTasa* : US\_Acquisition.fig

Tabla 4 : Métodos implementados en el archivo US\_Acquisicion.m

Ref	Funciones		Funciones invocadas	descripción
	Nombre	Comentarios		
	US_Acquisicion			
39	US_Acquisicion_OpeningFcn		XStreamML ShowImg	Inicia la ventana , fondos , iconos , botones ...
	US_Acquisicion_OutputFcn			No se utiliza
30	button_PausePrev_Callback	Callback del botón PAUSE ( <i>Post-record controls</i> )		Paraliza la reproducción del <i>preview</i>
31	button_PlayPrev_Callback	Callback del botón PLAY ( <i>Post-record controls</i> )	XStreamML.m	Visualiza la grabación capturada en la memoria de la cámara, <i>tWait</i> indica el tiempo de refresco.
29	button_StopRec_Callback	Callback del botón STOP ( <i>Record</i> )	XStreamML.m	Cancela la grabación
28	button_StartRec_Callback	Callback del botón REC ( <i>Record</i> )	XStreamML.m TimHubML.m initTimingHub	Comienza la grabación con los parámetros introducidos en el panel CAPTURE OPTION y configurados en las cámaras

40	Ret_Callback	Callback del botón CALIBRATE ( Controls )		Vuelve a la pantalla de alineamiento y ajuste
38	button_Acquire_Callback	Callback del botón ACQUIRE ( Controls )		Graba en disco las imágenes capturadas de la grabación
34 Bis	button_path_Callback	Callback del botón PATH ( Acquire options )		Indica la dirección de la carpeta donde se descargarán las imágenes
35	edit_FrameIni_Callback	Callback del edit FRAME INI ( Acquire options)		Indica el comienzo de las descarga
36	edit_FrameFin_Callback	Callback del edit FRAME FIN ( Acquire options)		Indica el final de la descarga
37	button_Eye_Callback	Callback del botón EYE ( Acquire options)	slider_preview_Callback	Visualiza la imagen que esta en el <i>edit</i> pegado a este botón.
34 Bis	button_Path_Callback	Callback del botón PATH ( Acquire options )		Indica la dirección de la carpeta donde se descargarán las imágenes
32	slider_preview_Callback	Callback del SLIDER ( Post - record controls )	XStreamML.m	Reproduce la captura y permite desplazar el control deslizante hasta el punto en el que desea visualizar
21	button_Exp_Ma_Callback	Callback del botón + f/2 ( Capture options )	Exposure_Callback	Aumenta la luminosidad ( exposure ) en un paso
22	button_Exp_Me_Callback	Callback del botón - f/2 ( Capture options )	Exposure_Callback	Disminuye la luminosidad ( exposure ) en un paso
23	Frames_Callback	Callback del edit NUMBER OF FRAMES ( Capture options)	XStreamML.m	Indica el numero de imágenes que se capturaran durante la grabación
19	Fps_Callback	Callback del edit FPS ( Capture options )	XStreamML.m , ShowImg	Indica la tasa de grabación de la captura
25	slider_trigger_Callback	Callback del SLIDER CIRCULAR REC ( Capture options )		Consultar en IDT
20	Exposure_Callback	Callback del edit EXPOSURE ( Capture options )	ShowImg , XStreamML.m	Actualiza el parámetro en las cámaras. Además controla que no se salga del margen
24	radio_RecCircular_Callback	Callback del radiobutton CIRCULAR ( Capture options)		Activa el panel con la barra de desplazamiento e información acerca del mismo
	ShowImg	Función similar a la implementada en US_Calibration>ShowImg	XStreamML.m , TimHubML.m , initTimingHub	Muestra imágenes de cada una de las cámaras conectadas , en caso de error devuelve un valor distinto de 0
	initTimingHub		TimHubML.m	Inicializa y configura el <i>timing hub</i> encargado de la sincronización , en caso de error devuelve un valor distinto de 0
33	sinc_Callback	Callback del botón SINC ( Post-record controls)		Sincroniza la reproducción de la grabación
27	Freez_Callback	Callback del botón FREZZED ( Controls)		Finaliza el modo LIVE

26	Live	Callback del botón LIVED ( <i>Controls</i> )	ShowImg	Inicia el modo LIVE
----	------	--	---------	---------------------

### 3.4.5. Panorámico

En la Ilustración 18 , zona derecha se presenta un diagrama con la relación entre los archivos utilizados para este modo. A continuación, se describen la funcionalidad de los archivos implementados.

Además, se explican las funciones implementadas para ejecutar y dotar de funcionalidad a las interfaces diseñadas para este modo: alineamiento y ajuste por un lado y por el otro, captura y montaje.

#### 3.4.5.1.DESCRIPCIÓN DEL SISTEMA DE ARCHIVOS

En la Tabla 5 están enumerados los archivos desarrollados para implementar este modo, además se incluyen algunos comentarios sobre los mismos. Las referencias mencionadas en los comentarios están relacionadas con la Ilustración 22.Los archivos marcados con \* son comunes a ambos modos.

Tabla 5 : Sistema de archivos para el modo *AumentoCampo*

Nombre	Comentarios
Pan_Calibration.fig	Interfaz Grafica asociado a Pan_Calibration.m
Pan_Calibration.m	
Pan_Acquisicion.fig	Interfaz Grafica asociado a Pan_Acquisition.m
Pan_Acquisicion.m	
sketch.fig	Interfaz Grafica asociado a sketch.m
sketch.m	Referencia 19 .Muestra un croquis de la escena donde se informa de la separación que deben tener entre las cámaras para poder obtener el resultado.
*XStreamML.dll	Extensión de la aplicación
*XStreamML.m	
*Menu_File_Exit_Callback.m	Referencia 3, funciones implementadas <b>CloseCameras</b> , comentada en <b>US_calibration</b> . Finaliza la aplicación
*Menu_File_Load_Callback.m	Referencia 1, funciones utilizadas <b>SendParameter2Cam</b> y <b>ShowImg</b> , comentadas en <b>US_calibration</b> . Esta función carga los parámetros de la configuración de las cámaras
*Menu_File_Save_Callback.m	Referencia 2 , esta función guarda los parámetros de configuración de las cámaras
*setIconImage.m	Función que crea el menú de herramientas
*CalcExposure.m	Función que calcula el tiempo de exposición mínimo y la estela borrosa
*CalcSep.m	Función que calcula de manera teórica la región común
*growRegion.m	Algoritmo de crecimiento de regiones
*Imagenes	Directorio con las imágenes de fondos de la aplicación
*icons	Directorio de iconos de la barra de herramientas y reproductor



3.4.5.2.DIAGRAMA DEL ASISTENTE DE ALINEAMIENTO Y AJUSTE

A continuación solo se describen las funciones añadidas al modo *AumentoTasa* para implementar este modo; el resto son idénticas a las del modo *AumentoTasa*. Las funciones, marcadas con el símbolo \*, están explicadas en la Tabla 3 .

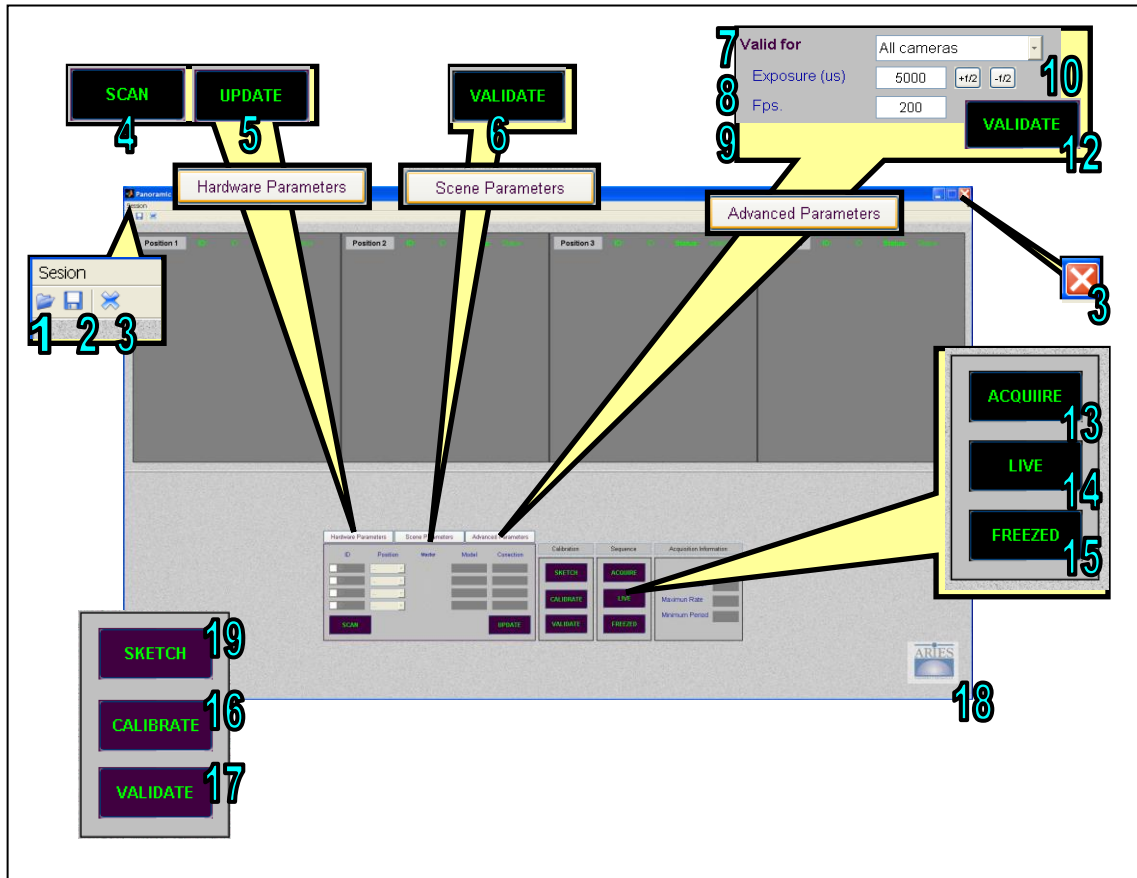


Ilustración 22 : Interfaz de alineamiento y ajuste del modo *AumentoCampo*: Pan\_Calibration.fig

Tabla 6 Métodos implementados en el archivo Pan\_Calibration.m

Ref	Funciones		Funciones invocadas	Descripción
	Nombre	Comentarios		
18	Pan_Calibration		InitStruct ShowImg SenParameters2 Cam OpenCameras setIconImage.m XStreamML.m	Comienza la aplicación , se inicializan las variables y se detectan las cámaras conectadas
	Pan_Calibration_OpeningFcn		CloseCameras	Finaliza la aplicaron , se cierran las cámaras conectadas
	Pan_Calibration_OutputFcn			
4	*Scan_Callback			
5	*Update_Callback			
6	*UpdateScene_Callback			
7	*CameraldAdvanced_Callback			

12	*UpdateAdvanced_ Callback			
10	*Ma_Exposure_Callback			
10	*Me_Exposure_Callback			
8	*Exposure_Callback			
9	*Fps_Callback			
14	*Play_Callback			
13	*Acquire_Callback			
15	*Stop_Callback			
	*InitStruct			
	*CloseCameras			
	ShowImg			Presenta cambios con respecto a su homóloga en el otro modo. No se visualiza el 10% del margen de la imagen. Se utiliza para la mejora de la calidad .
	*OpenCameras			
	*SendParameter2Cam			
	MasterSelect_Callback	Callback del desplegable Master Select_Callback		Selecciona la cámara que será establecida como MASTER del sistema para la sincronización de las cámaras
16	CalEnd_Callback			Inicia el algoritmo de alineamiento, para la marcación de puntos.
17	Cal_Callback			
	calcMovCamera			Calcula el desplazamiento entre dos imágenes, solapadas. Resultado del alineamiento

### 3.4.5.3. DIAGRAMA DE LA HERRAMIENTA DE CAPTURA Y MONTAJE

Al igual que en el apartado anterior, tan solo se añaden comentarios a las funciones que han sido añadidas con respecto al modo *AumentoTasa*. El resto de funciones, marcadas con el símbolo \*, están explicadas en la Tabla 4.

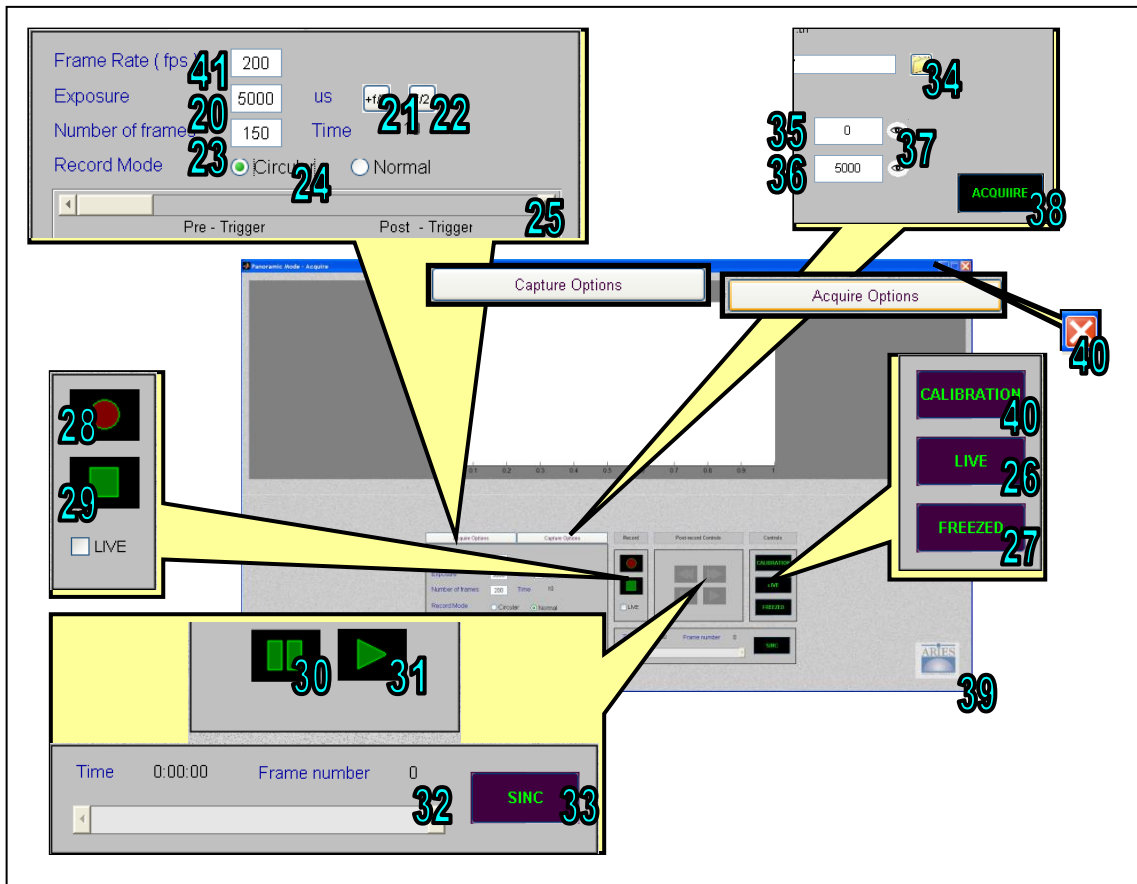


Ilustración 23: Interfaz de Captura y montaje del modo AumentoCampo : Pan\_Acquisition.fig

Tabla 7: Métodos implementados en el archivo Pan\_Acquisicion.m

Ref	Funciones		Funciones invocadas	Descripcion
	Nombre	Comentarios		
39	Pan_Acquisicion			
	Pan_Acquisicion_OpeningFcn			
	Pan_Acquisicion_OutputFcn			
30	*button_PausePrev_Callback			
31	*button_PlayPrev_Callback			
29	*button_StopRec_Callback			
28	button_StartRec_Callback			En esta función, se modifica el modo de grabación de las cámaras para que la cámara MASTER aporte el sincronismo al sistema y el resto la reciba , ESLAVE. El resto es idéntico al modo <i>AumentoTasa</i>
40	*Ret_Callback			
38	*button_Adquire_Callback			
35	*edit_FrameIni_Callback			

36	*edit_FrameFin_Callback			
37	*button_Eye_Callback			
34	*button_Path_Callback			
32	*slider_preview_Callback			
21	*button_Exp_Ma_Callback			
22	*button_Exp_Me_Callback			
23	*Frames_Callback			
41	*Fps_Callback			
25	*slicer_trigger_Callback			
20	*Exposure_Callback			
24	*radio_RecCircular_Callback			
	ShowImg	Utiliza la función 'SynchGrab' de la interfaz de la cámaras	XStreamML.m	Muestra imagen panorámica compuesta a partir de las imágenes individuales de cada cámara para el modo LIVE
	calcSizeFrame			Esta función calcula el tamaño del objeto <i>frame</i> para la configuración de las cámaras determinada
	ShowImgRec	utiliza la función 'MemoryReadData' de la interfaz de la cámaras	XStreamML.m	Muestra imagen compuesta , después de la grabación
27	*Freez_Callback			
26	*Live			

### 3.4.6. Cinematic Analysis

En este apartado se describe el sistema de archivos, diagrama funcional y una descripción de las funciones implementadas para desarrollar esta interfaz. El propósito de esta aplicación es generar resultados tangibles a partir de puntos marcados manualmente en las imágenes de la secuencia. Por lo tanto, con respecto a las anteriores , se elimina la dependencia del sistema.

#### 3.4.6.1.DESCRIPCIÓN DEL SISTEMA DE ARCHIVOS

En la Tabla 8 , se describen los archivos desarrollados. Estos archivos están relacionados según la Ilustración 19 , donde se representa las relaciones utilizando un diagrama de flujo.

Tabla 8 : Sistema de archivos para la aplicación de cálculo de medidas, *Cinematic Analysis*

Nombre	Comentarios
--------	-------------

<b>angle3Points.m</b>	Este archivo contiene una función que calcula en Angulo en radianes de 3 puntos que se le pasan como argumento. Cortesía de <i>Matlab file interchange</i>
<b>Distance.fig</b>	Interfaz grafico asociado a Distance.m
<b>Distance.m</b>	Esta ventana aparece para solicitar al usuario que introduzca una distancia en centímetros en un objeto tipo edit , además de botón de aceptar y cancelar
<b>Fase3.fig</b>	Interfaz grafico asociado a Fase3.m
<b>Fase3.m</b>	
<b>findNextFile.m</b>	Función que busca el siguiente archivo de la secuencia , suponiendo que estos están ordenados de manera alfa-numérica
<b>findPrevtFile.m</b>	Función que busca el siguiente archivo de la secuencia , suponiendo que estos están ordenados de manera alfa-numérica
<b>getFps.fig</b>	Interfaz grafica asociado a getFps.m , Ilustración 73
<b>getFps.m</b>	La funcionalidad es la misma que Distance, Tan solo solicita otro tipo de datos , el numero de Fps .
<b>getZoom.fig</b>	Interfaz asociado a getZoom.m
<b>getZoom.m</b>	Función creada para hacer un zoom del punto que se le pasa como argumento sobre la imagen. Devuelve el punto marcado en la imagen , en una variable del objeto axes denominada UserData . La magnitud del zoom se controla mediante la variable handles.zoom . La dimensión de la ventana esta controlado por el diseño de la interfaz en getZoom.fig . No bloquea la ejecución de la aplicación puesto que ejecuta una rutina cuando se produce un evento , clic , sobre la ventana .
<b>Icons</b>	Directorio que contiene los iconos utilizados
<b>zoomImg.m</b>	Función creada para hacer un zoom en un punto sobre una imagen que se pasan como argumentos. A diferencia de getZoom.m, esta función es mas genérica y podría ser utilizada para otras necesidades. La función crea un marco con unas dimensiones controladas por dos parámetros: tamaño de zoom en píxeles (sizeZ ) y numero de píxeles en la ventana por píxel real ( magZ ) . Devuelve como argumento el punto marcado sobre la imagen con origen de coordenadas en la imagen total. Esta función se utiliza para realizar la calibración de la secuencia, necesario para generar los resultados, cuando se solicita al usuario que marque una línea en la imagen e indique la longitud real de la misma . Bloquea la ejecución de la aplicación hasta que se produce un evento, clic , sobre la ventana .

### 3.4.6.2.DIAGRAMA FUNCIONAL

En este apartado se describen las funciones de la aplicación desarrollada. La Ilustración 24 , muestra un aspecto detallado de cada uno de los botones de esta aplicación. La tabla siguiente relaciona cada uno de los objetos numerados con sus respectivas funciones y algunos detalles de las mismas.

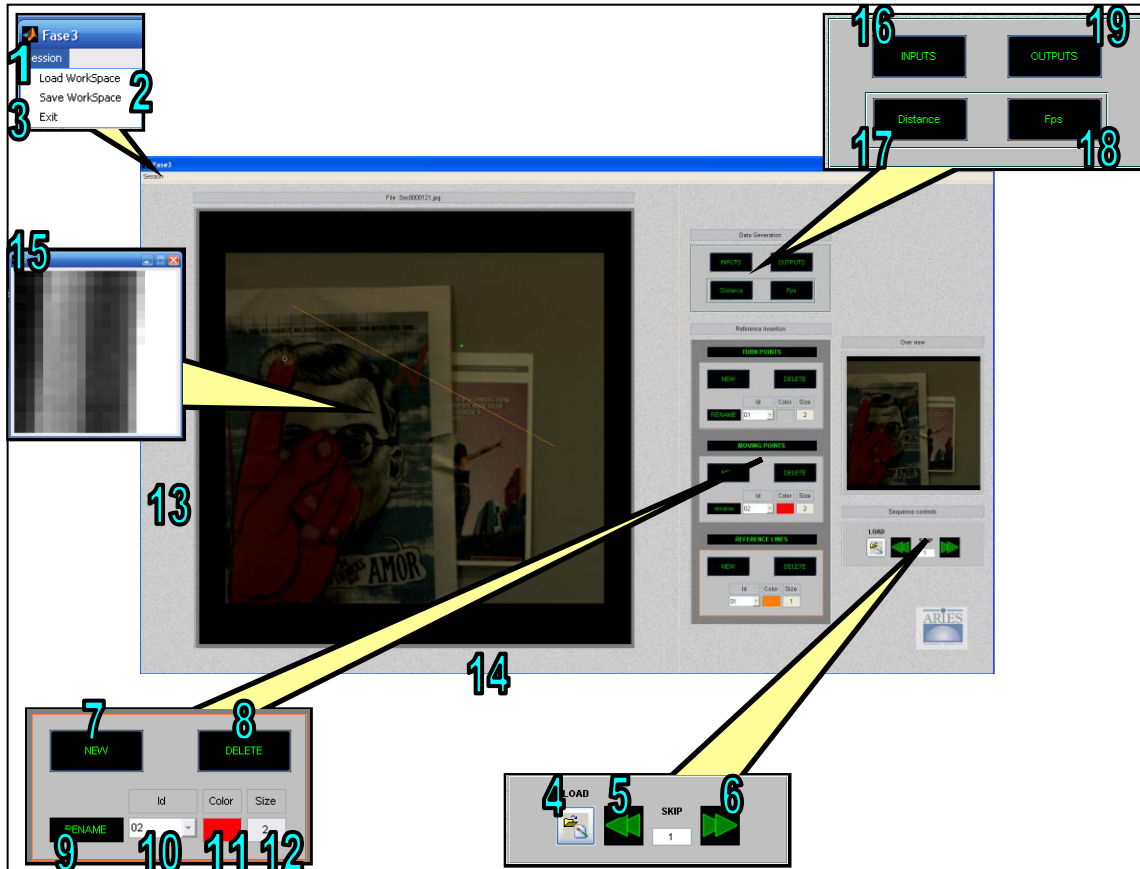


Ilustración 24 : Interfaz de *Cinematic Analysis*: Fase3.fig

Tabla 9 :Métodos implementados en el archivo Fase3.m

Ref	Funciones		Funciones invocadas	Descripción
	Nombre	Comentarios		
	Fase3			
	Fase3_OpeningFcn	Se ejecuta antes de que la interfaz se haga visible		Carga iconos , fondos , e inicializa el numero de objetos máximo que se podrán crear
	Fase3_OutputFcn	No se utiliza		
13	slider1_Callback	Callback del SLIDER horizontal	showImg	Con esta función permite desplazarse verticalmente por la imagen que no se pueden visualizar enteras en el objeto axes principal.
14	slider2_Callback	Callback del SLIDER vertical	showImg	Con esta función permite desplazarse horizontalmente por la imagen que no se pueden visualizar enteras en el objeto axes principal.
4	LoadImg_Callback	Callback de botón LOAD	initArrays ,showImg	Esta función permite seleccionar un archivo-imagen, que pertenezca a una secuencia de imágenes. También inicializa las estructuras que contendrán los datos de los puntos , líneas y puntos de giro marcados en cada una de las imágenes
5	button_Prev_Callback	Callback del botón PREV del panel <i>sequence controls</i>	showImg, showObjects, findPrevFile.m	Utiliza la función findPrevFile.m para obtener el siguiente fichero de la secuencia. Muestra esta imagen en la pantalla junto con los objetos ( puntos , líneas y puntos de giro )

				asociados
6	<b>button_Next_Callback</b>	Callback del botón NEXT del panel <i>sequence controls</i>	<b>showImg, showObjects, findNextFile.m</b>	Utiliza la función <code>findNextFile.m</code> para obtener el siguiente fichero de la secuencia. Muestra esta imagen en la pantalla junto con los objetos ( puntos , líneas y puntos de giro ) asociados
	<b>showObjects</b>		<b>findIdxFileInDir</b>	Muestra los objetos (puntos, líneas y puntos de giro) asociados a la imagen que esta en pantalla . Para ello recurre a las estructuras creadas para tal efecto .
	<b>showImg</b>	Devuelve dos imágenes, una para cada axes del la ventana. Ambas deben ser cuadradas por lo que se utiliza <i>squareImg</i>	<b>squareImg</b>	Devuelve imágenes cuadradas para cada axes, además habilita los slider si resultan necesarios.
8	<b>delLine_Callback</b>	Callback del botón DELETE del panel line		Oculto la línea seleccionada
8	<b>delTurn_Callback</b>	Callback del botón DELETE del panel turn		Oculto el punto de giro seleccionado
8	<b>delPoint_Callback</b>	Callback del botón DELETE del panel points	<b>showObjects, showImg</b>	Oculto el punto seleccionado , ademas refresca la imagen
7	<b>newLine_Callback</b>	Call-back del botón NEW del panel Line		Crea un nuevo objeto tipo <i>line</i> y lo añade a la estructura de datos ArrayL
7	<b>newTurnPoint_Callback</b>	Call-back del botón NEW del panel Turn		Crea un nuevo objeto tipo <i>rectangle</i> y lo añade a la estructura de datos ArrayT
7	<b>newPoint_Callback</b>	Call-back del botón NEW del panel point		Crea un nuevo objeto tipo <i>rectangle</i> y lo añade a la estructura de datos ArrayP
10	<b>IdLine_Callback</b>	Callback del desplegable ID del panel <i>Line</i>		Permite seleccionar el punto de giro que se va a utilizar, ya bien para ubicarlo sobre la imagen, modificar su tamaño, color, nombre.
10	<b>IdTurn_Callback</b>	Callback del desplegable ID del panel <i>Turn</i>		Permite seleccionar la línea que se va a utilizar, ya bien para ubicarlo sobre la imagen o modificar su tamaño, color .
10	<b>IdPoint_Callback</b>	Callback del desplegable ID del panel <i>Points</i>		Permite seleccionar el punto que se va a utilizar, ya bien para ubicarlo sobre la imagen, modificar su tamaño, color, nombre .
11	<b>Color_Callback</b>	Callbacks de los botones COLOR	<b>setButtonColor ,pushPoint ,HighLightPanel</b>	Se utiliza para seleccionar un color y aplicarlo sobre el objeto que llamo a la función, punto , línea o punto de giro .
	<b>initArrays</b>			Crea estructura de datos con el nombre de los archivos que se pasan como argumento, estructura proveniente del comando dir, que contendrá las coordenadas y características de los objetos: punto, líneas , puntos de giro .
	<b>findIdxFileInDir</b>			Localiza la posición, índice, de la imagen actual dentro de la estructura de datos manejada .
	<b>push</b>		<b>getZoom.m</b>	Se ejecuta cuando se pulsa sobre el axes principal. Luego , dependiendo del tipo de objeto seleccionado , permite situar una línea marcando dos puntos sobre la imagen o bien invoca a la función <code>getZoom.m</code> para que haga un zoom y marcar un punto de manera precisa
	<b>pushPoint</b>	Callback del axes principal	<b>findIdxFileInDir, showObjects, showImg</b>	Esta función puede ser ejecutada de dos maneras: desde la función <code>getZoom.m</code> o desde funciones internas, p.ej. <code>Color_callback</code> . De la primera forma y mediante la variable <i>push</i> , se indica que un punto nuevo ha sido marcado proviene de la

				función getZoom, se actualizan las coordenadas en la estructura. Se refresca la imagen y los objetos para ambas modalidades .
	<b>calcPointAx</b>			Esta función devuelve la posición actual del cursor con respecto al axes principal, se utiliza a la hora de marcar líneas en la imagen.
	<b>pushCalibrate</b>	Callback del axes principal, y previamente a sido pulsado el botón calibrate.	<b>zoomImg.m ,calcPointAx ,Distance</b>	Utilizando zoom para una mayor precisión , se usa para marcar una línea recta sobre la imagen principal y a continuación indicar el tamaño real en cm. Calcula la relación entre píxel y tamaño real
16	<b>Calibrate_ Callback</b>	Callback del botón INPUTS		Muestra una ventana con los procedimientos para realizar la calibración y poder calcular los resultados
17	<b>InsDistance_ Callback</b>	Callback del botón DISTANCE		Muestra una ventana con los procedimientos para insertar la variable distancia , cambia el callback del axes principal de pushPoint a pushCalibrate
18	<b>InsFps_ Callback</b>	Callback del boton FPS	<b>getFps</b>	Permite introducir el valor de Fps de la secuencia , necesario para obtener los resultados
19	<b>calcResults_ Callback</b>	Callback del boton OUTPUTS	<b>initResult,initStruct,saveWorkSpace,</b>	Genera los resultados y los exporta a un archivo .xls , legible por EXCEL .
	<b>initResult</b>			Devuelve una estructura creada para almacenar los cálculos realizado en la función que la invoca
	<b>initResultTurn</b>			Devuelve una estructura creada para almacenar los cálculos realizado en la función que la invoca
9	<b>renamePoint_ Callback</b>	Callback del boton RENAME del panel POINTS		Permite modificar el identificador asignado al punto actual, luego se vera reflejado en las distintas hojas de datos generadas .
9	<b>renameTurnP_ Callback</b>	Callback del boton RENAME del panel TURN		Permite modificar el identificador asignado al punto de giro actual, luego se vera reflejado en las distintas hojas de datos generadas .
	<b>initStruct</b>			Inicializa la estructura que contendrá los datos generados para los puntos de giro : velocidad angular , aceleración angular ...
1	<b>saveWorkSpace</b>	Callback del SAVE del menú de Opciones		Salva el estado actual de la aplicación , lo guarda en una extensión tipo .dat que elige el usuario
2	<b>loadWorkSpace</b>	Callback del LOAD del menú de Opciones		Recupera un análisis previamente recuperado , el archivo debe tener la extensión .dat
3	<b>Exit_ Callback</b>	Callback del EXIT de menú Opciones		Finaliza la aplicación , antes pregunta al usuario si desea guardar el espacio de trabajo
	<b>updateObjects</b>			Esta función esta implementada para refrescar los objetos en las imágenes , muestra las líneas , puntos y puntos de giro
	<b>squareImg</b>			Devuelve una imagen cuadrada ( mismo alto que ancho) de una imagen que se pasa como argumento
	<b>HighLighPanel</b>			Colorea con un borde anaranjado el panel correspondiente que se le pasa como argumento
	<b>setButtonColor</b>			Colorea con el color que se pasa como argumento, el botón que invoca esta función .



### 3.5. Problemas y trabajo futuro

Muchos fueron los problemas que surgieron mientras se desarrollaban estas aplicaciones; unos llevaron mas tiempo que otros en ser solucionados, otros se tuvo que replantear la solución y otros no se encontró solución. A continuación, se describen algunos de los más relevantes:

- Las primeras versiones de los *drivers* y por extensión del SDK de las cámaras no pueden manejar múltiples cámaras.
- Cuando se produce un *trigger* vía hardware, no es posible conocer cuando ocurrió este evento. Para el modo *AumentoCampo* no resulta necesario pues todas las cámaras graban simultáneamente. En el modo *AumentoTasa*, resulta necesario conocer que cámara comenzó a grabar primero para conocer en que orden se deben intercalar las imágenes de cada una de las cámaras y conseguir la secuencia resultante. (Solución: Botón SYNC – El apartado A.4.3 en el panel POST RECORD CONTROLS se explica más en profundidad).
- La continúa revisión de la interfaz de usuario, elección de colores y fondos que dieran a la aplicación un aspecto de producto comercial profesional.
- El redimensionamiento de la ventana, un continuo problema cada vez que se modifica la resolución. Esta aplicación fue desarrollada para ejecutarse con una resolución de 1920 x 1200. En cualquier otra resolución aparece la ventana con sus objetos modificada.
- La aplicación debía soportar cámaras de varios modelos, con distintas resoluciones y monocromática y a color. La aplicación originalmente se ideó para que funcionara con el modelo X4, monocromática y con una resolución máxima de 512 x 512. La utilización de un nuevo modelo para montar el sistema, Y3, con resolución 1280x1024 a color produjo varias semanas de trabajo para adaptarlo a la aplicación.
- La conexión vía GigabitEthernet. La longitud de los cables USB esta limitada a unos 15 metros. Para montar el modo *AumentoCampo* se necesita en algunos casos que las cámaras estén separadas más de 15 metros, esto obliga a que sea Ethernet la tecnología utilizada para la interconexión de las cámaras con la aplicación.
- Para futuros trabajo sobre estas interfaces se propone:
  - Unificar las aplicaciones de los modos *AumentoTasa* y *AumentoCampo* en una única ventana donde se adaptara la ubicación de los objetos al modo de operación de manera automática. También seria interesante unificar la etapa de alineamiento y ajuste con la etapa de captura y montaje.
  - Implementación de algún método que permitiera el redimensionamiento automático de la ventana.

### 3.6. Conclusiones

Como resumen, se justifica la necesidad de desarrollar, utilizando MATLAB como lenguaje de programación, unas aplicaciones propias para el manejo y control del sistema. Para cada modo, *AumentoTasa* (aumento de la resolución temporal) y *AumentoCampo* (aumento de la resolución espacial), se desarrolló una interfaz para el alineamiento y ajuste y otra para la captura y montaje de la secuencia resultante.

Además de estas herramientas se implemento una interfaz para el estudio de las imágenes obtenidas en la fase anterior. En esta aplicación, mediante la marcación manual de puntos y puntos de giro, se realiza una estimación de algunas magnitudes físicas tales como velocidad, aceleración... también se puede comprobar la calidad y la importancia de las mejoras realizadas a la secuencia resultante.

Hay que tener en cuenta que para realizar estas estimaciones es necesario conocer algunas características de la secuencia estudiada: tasa de grabación de la secuencia y alguna distancia real de algún objeto que aparezca en la secuencia. Estas dos variables resultan fundamentales para conseguir unas estimaciones precisas de las magnitudes estimadas, al igual que la precisión en la marcación de los puntos de referencia. El resultado de este estudio se plasma sobre un documento *.xls* , legible por *MS Excel*.

Para el desarrollo de estas aplicaciones se utilizo el *toolbox* de MATLAB que facilita la creación de interfaces, GUIDE. Para todas las aplicaciones, existe un archivo con extensión ‘.fig’ que representa la interfaz . Esta contiene la definición de los objetos, su ubicación y características. Como archivo relacionado a la interfaz sin el cual la aplicación no puede funcionar, GUIDE crea un archivo con el mismo nombre que la interfaz pero con extensión ‘.m’, donde se desarrollan todas las funciones: *callbacks* como respuesta a un evento ocurrido sobre un objeto de la interfaz y otros métodos auxiliares.

## 4. Desarrollo de algoritmos de corrección de los errores asociados al proceso de captación y fusión

---

En este capítulo se abordan los conceptos utilizados para implementar las herramientas necesarias para conseguir una mejora en la calidad de la secuencia final. Las secuencias de las que se parte han sido capturadas previamente por un sistema de cámaras dispuestas según el objetivo deseado: aumento de la resolución espacial o temporal, descrito en el capítulo 2

Además, se analizará el resultado de la fusión e intercalación de las imágenes de cada cámara y las herramientas desarrolladas para incrementar la calidad. Como se ha hecho hasta este punto, serán separados los dos modos de operación: aumento de la resolución temporal, modo *AumentoTasa*, y aumento de la resolución espacial, modo *AumentoCampo*.

En cada uno de los modos existen unas limitaciones del sistema que producen que la secuencia resultante no sea óptima. Estas limitaciones y los problemas que generan serán analizados a continuación.

Antes de continuar, hay que comprender el proceso de captura de una cámara para justificar algunas transformaciones realizadas. Una imagen es una representación en dos dimensiones (imagen) del espacio de tres dimensiones (escena). En el modelo más simple de cámara denominado *pinhole* (ver Ilustración 25), un rayo de luz de un punto en el espacio, atraviesa la lente, punto focal y se plasma sobre la película o dispositivo digital (sensor), produciendo una imagen del punto. En este modelo, el centro de la lente representa un punto en el espacio que todos los rayos atraviesan para formar la imagen. Y la imagen es el plano donde se proyectan los rayos, denominado plano de proyección o plano de imagen.

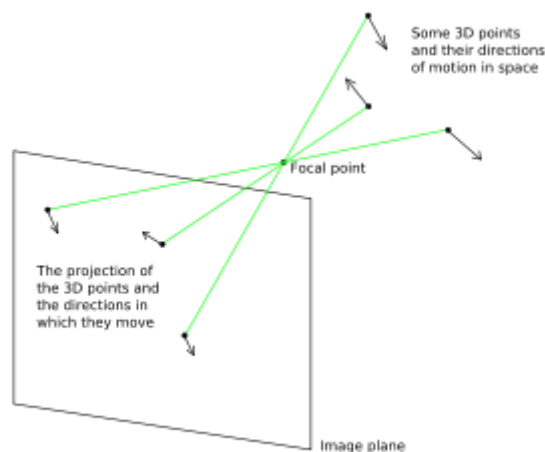


Ilustración 25: Modelo *pinhole*

### 4.1. Requisitos iniciales

#### 4.Desarrollo de algoritmos de corrección de los errores asociados al proceso de captación y fusión

Esta etapa se orienta al desarrollo de algoritmos de corrección de los errores asociados al proceso de captación y fusión. El objetivo de esta fase es añadir a la aplicación la funcionalidad necesaria para mejorar la calidad de la secuencia final. Esta sección enumera las técnicas que, tras un análisis inicial del tipo de secuencias que se van a obtener, se considera de aplicación imprescindible para lograr que en la secuencia final sea inapreciable el hecho de que haya sido captada por más de una cámara.

Conviene tener en cuenta que las correcciones previstas tienen como objetivo mejorar la calidad final desde el punto de vista de la precisión de las medidas que posteriormente se realicen sobre los objetos en movimiento. No se trata, por lo tanto, de simples ajustes estéticos, ampliamente utilizados en programas de retoque de imágenes fotográficas, que en raras ocasiones preservan la geometría de la escena.

### **4.2. Modo AumentoTasa**

Con un sistema de cámaras, más de dos, es posible aumentar la resolución temporal de la secuencia resultante tantas veces como cámaras se utilicen. La tasa de grabación debe ser la misma para todas, además de estar sincronizadas. En la práctica, el sistema controla el sincronismo de las cámaras para que capturen en tiempos distintos y sea posible conseguir el objetivo. La secuencia final, se obtiene ordenando las imágenes capturadas por cada una de las cámaras del sistema. Para más detalles consultar el capítulo 2.

Una vez obtenida la secuencia final, resultado de la intercalación de imágenes de cada una de las cámaras, se puede observar en la secuencia resultante los siguientes efectos que deterioran la calidad subjetiva de la secuencia final:

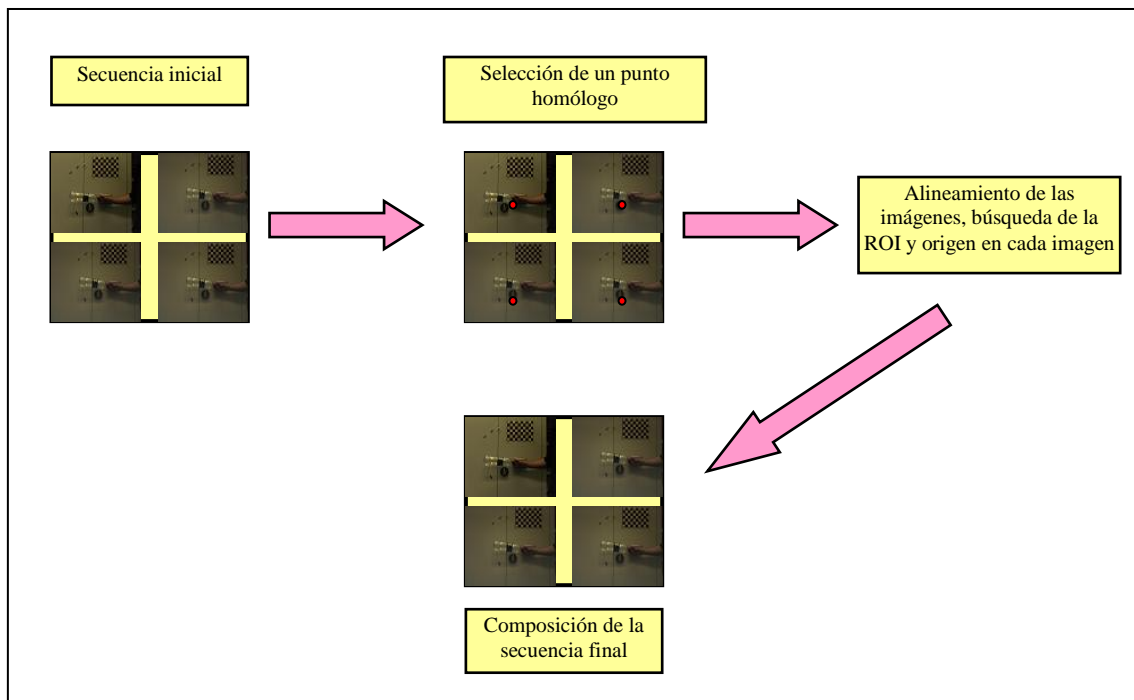
- Debido a la óptica utilizada, enfoque e iluminación. Existen diferencias de contraste y enfoque entre las imágenes de cada una de las cámaras. Cuando se reproduce la secuencia final este problema origina un efecto de *parpadeo* entre imágenes consecutivas, además de contornos con diferente realce debido a la diferencias de enfoque.
- Debido a que las cámaras no capturan desde el mismo punto de vista, objetos que están estáticos en la escena real se observan alternando su posición en la secuencia resultante. Este fenómeno es más acusado cuando aparecen distintos objetos colocados en planos muy alejados del plano de objeto (aquel en el que se sitúa el objeto bajo observación, y sobre el que se implementa el algoritmo de alineamiento).
- Otros fenómenos: los relativos a posibles desviaciones de los parámetros intrínsecos de la cámara, fenómenos que en este PFC no serán abordados (las tolerancias ofrecidas por el fabricante se consideran suficientes para la aplicación bajo estudio); y los relativos al efecto de las lentes (distorsión radial y tangencial).

Estos problemas pueden minimizarse actuando sobre los parámetros de captura. Por un lado siendo muy preciso al ajustar las ópticas a iguales valores, tanto enfoque como apertura del obturador. Además se intenta, en la medida de lo posible, que las

#### 4.Desarrollo de algoritmos de corrección de los errores asociados al proceso de captación y fusión

cámaras captan desde una perspectiva lo más parecida posible: juntando las cámaras todo lo que su estructura física permite y capturando escenas que se desarrollen en un único plano y que estén muy alejadas del sistema. Aún así, en la práctica aparecen diferencias de contraste en objetos y zonas fijas que se mueven debido a la diferencia de perspectiva. Por ello es necesario corregir la secuencia directamente generada por las aplicaciones descritas en el capítulo anterior.

En una primera aproximación al problema, se calculó la región visible común (ROI *region of interest*) a todas las cámaras y se recortó este área en cada imagen, según se desarrolla en el apartado 4.4



**Ilustración 26: Esquema de la composición de la secuencia para el modo *AumentoTasa***

Observando la secuencia obtenida con esta solución, se evidenciaron todos los defectos citados anteriormente. Sin embargo, se consideró válida como punto de partida y como medio para evaluar la utilidad de las futuras mejoras implementadas .

La Ilustración 27, muestra un diagrama de bloques que representa las operaciones y transformaciones realizadas sobre la secuencia inicial que consiguen que la secuencia final resulte mejorada cuantitativa y cualitativamente.

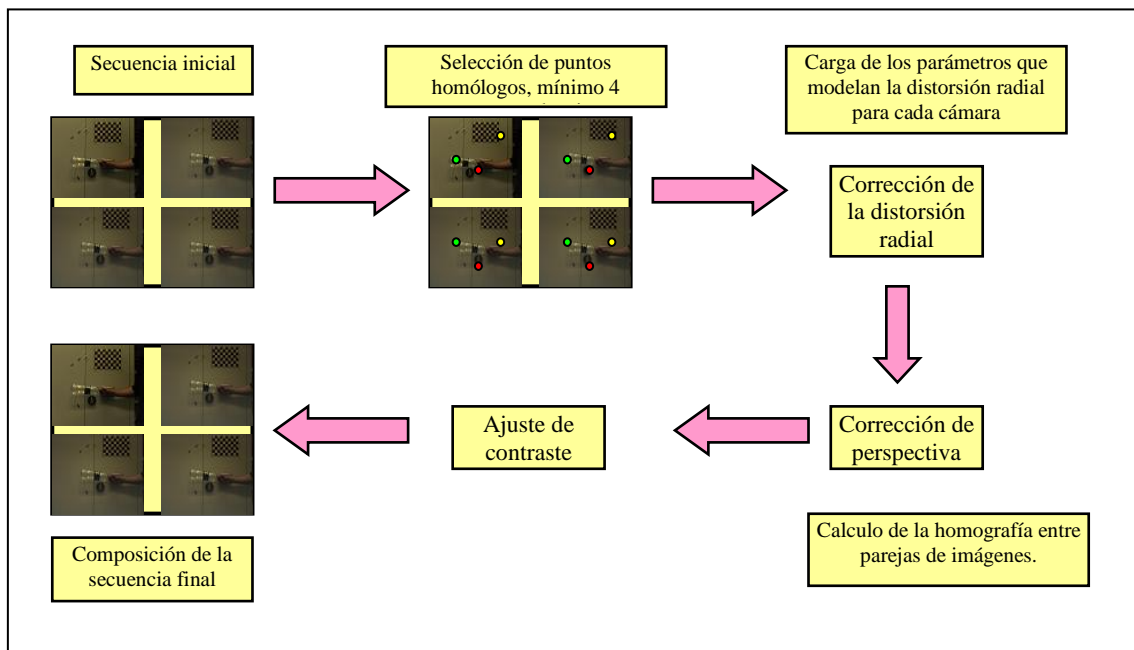
En el proceso de captura, las lentes utilizadas no se comportan como en el modelo ideal. Debido a las técnicas de producción y materiales utilizados las lentes introducen aberraciones que distorsionan la escena representada en la imagen. La distorsión más predominante es la distorsión radial que produce que las líneas rectas en la escena real no sean representadas como tales en la imagen. Esta distorsión depende de la lente utilizada así como de la ubicación de la misma con respecto al sensor. Aprovechando que las ópticas utilizadas eran de distancia focal fija, se realizaron trabajos para intentar atenuar esta distorsión.

#### 4.Desarrollo de algoritmos de corrección de los errores asociados al proceso de captación y fusión

En un sistema de 2 cámaras, las imágenes impares de la secuencia resultante han sido capturadas desde un punto de vista (cámara 1) y las impares desde otro (cámara 2). Asumiendo que la escena se desarrolla en un plano, es posible encontrar una transformación sobre el plano proyectivo que relaciona estos puntos de vista, sin conocimiento adicional sobre la composición de la escena y utilizando únicamente las imágenes de cada una de las cámaras. Esta transformación se obtiene a partir de correspondencias de puntos entre cada plano proyectivo, imagen de la cámara 1 con otra de la cámara 2, y permite unificar el punto de vista de dos imágenes. Aplicando esta transformación a las imágenes capturadas por la cámara 2, se consigue que estas imágenes sean representadas como si hubieran sido obtenidas por la cámara 1. Este razonamiento es extensible a cualquier número de cámaras siempre que la transformación se haga utilizando la misma cámara como referencia.

Por último, además de las correcciones citadas, resulta evidente la necesidad de hacer un ajuste de contraste para reducir las diferencias de intensidad de cada píxel con respecto a la imagen anterior y posterior, que corresponden a cámaras distintas. El ajuste de contraste se realiza por separado entre dos zonas de la imagen denominadas fondo y objeto. Este ajuste genera una pérdida de resolución en el fondo de las imágenes de la secuencia resultante del proceso, pérdida que no debe afectar al objeto móvil.

Las herramientas y transformaciones que aquí se exponen (corrección de distorsión radial, corrección de perspectiva y ajuste de contraste) han sido desarrolladas e implementadas según se describe en este documento. En apartados posteriores se abordaran las bases matemáticas que modelan cada herramienta así como la algoritmia y archivos implementados para la consecución del objetivo.



**Ilustración 27: Esquema de las mejoras propuesta para mejorar la secuencia final en el modo US**

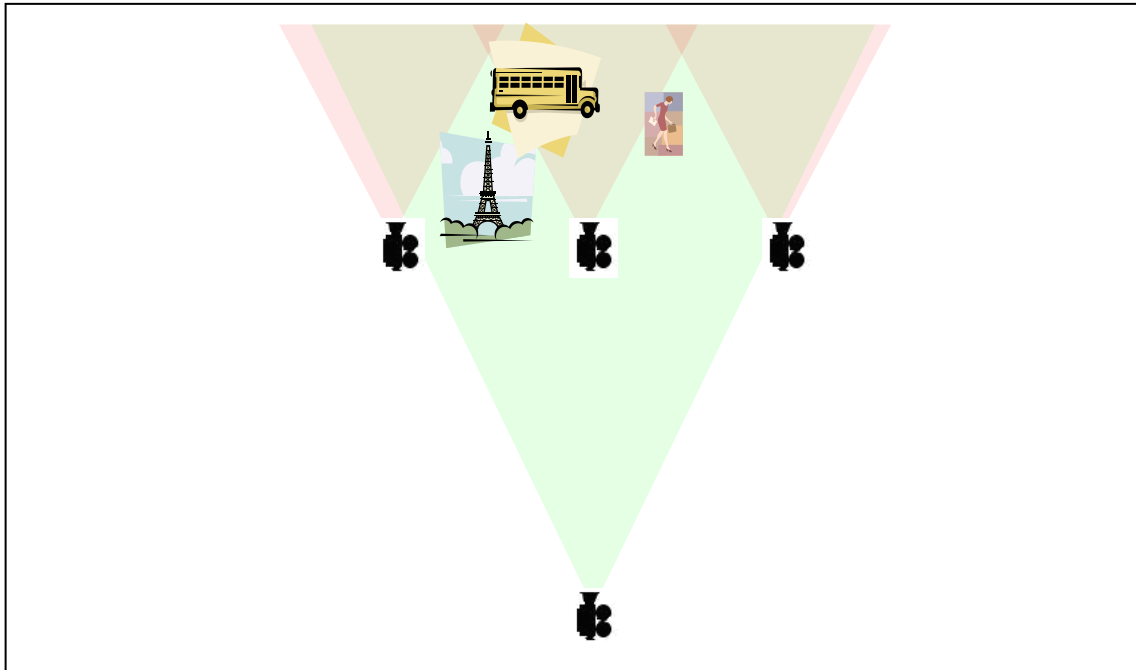
### **4.3. Modo AumentoCampo**

En este modo de operación, las imágenes capturadas por cada una de las cámaras en el mismo instante de tiempo, se registran (fusionan) para conseguir una imagen panorámica. Para poder conseguirlo, todas las cámaras deben estar sincronizadas de manera que capturen en sincronía, además de estar separadas una distancia y paralelas al plano que se quiere capturar. Con esto, se desea simular que la secuencia final ha sido capturada desde un único punto de vista que está ubicado más lejos de lo que lo está el sistema con respecto a la escena capturada. De esta manera se obtiene una secuencia final panorámica con una resolución en una de las dimensiones proporcional al número de cámaras utilizadas en el sistema.

En este modo, la secuencia final se consigue fusionando las imágenes capturadas por las cámaras en el mismo instante. Para realizar la fusión se necesita una región común en las imágenes para que pueda encontrarse una correspondencia. Analizando el proceso de fusión sin ninguna corrección, se observan distintos fenómenos que deterioran la calidad de la secuencia final:

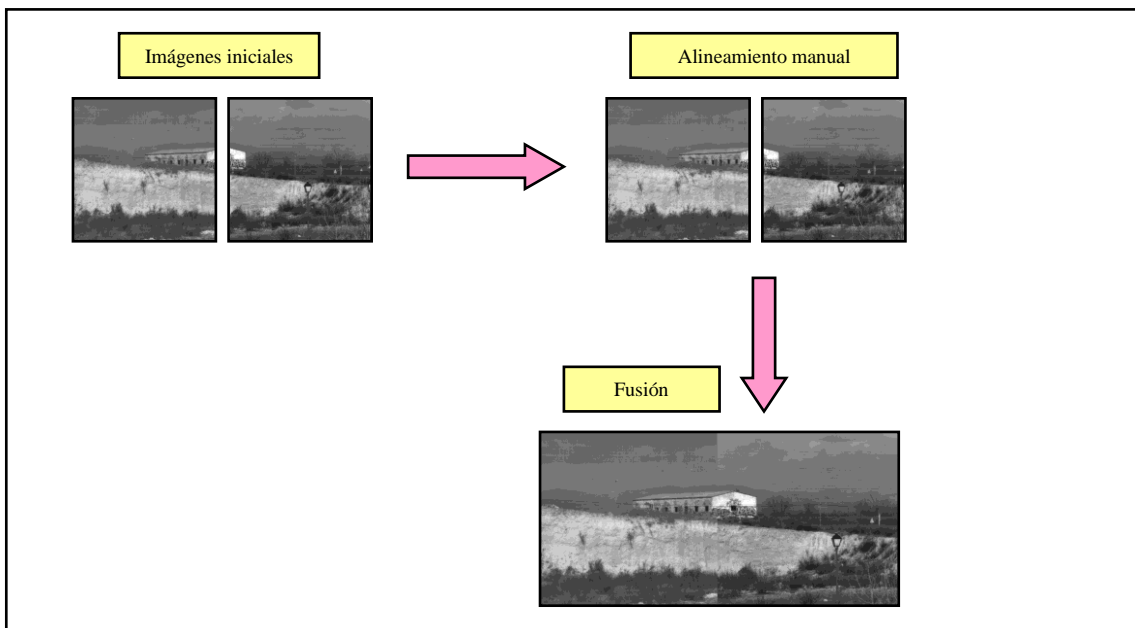
- Al igual que en modo anterior, aparece una diferencia de contraste y enfoque debido a las diferencias entre ópticas utilizadas y características del sensor de cada cámara.
- Además de esta diferencia de contraste, la óptica utilizada introduce distorsión radial. Aunque en las lentes utilizadas es poco perceptible, ésta aumenta a medida que uno se aleja del centro de distorsión (que habitualmente está en el centro de la imagen). Esto afecta en mayor medida a la búsqueda de una correspondencia ya que las zonas comunes a dos imágenes se encuentran precisamente en los extremos de las éstas.
- Otros problemas son debidos a la situación de las cámaras. En la Ilustración 28, podemos visualizar este problema. El objetivo de esta fusión es simular que una cámara panorámica ubicada en un punto más lejano a nuestro sistema. Imaginemos una escena donde queremos capturar un autobús que transcurre por una carretera, éste plano es paralelo al sistema de cámaras. Si colocamos las cámaras de una determinada manera, podremos realizar la fusión de las imágenes y conseguir continuidad. Ahora imaginemos que en un plano mas próximo pasa una chica caminando, esta mujer aparecerá en las imágenes sólo cuando se encuentre dentro del campo de visión de alguna de las cámaras; por lo tanto no tendrá continuidad y nuestro objetivo no será conseguido.
- A la hora de componer la imagen panorámica, resulta útil que las cámaras se encuentren más o menos alineadas horizontal y verticalmente entre sí. Aunque en los requisitos iniciales este alineamiento se pedía que fuera automático, analizando distintas soluciones, se optó por implementar un asistente de alineamiento que ayudara al usuario a ubicar las cámaras. De este aspecto, el alineamiento, depende en gran parte la calidad de las mejoras y el resultado de la composición de la panorámica

#### 4.Desarrollo de algoritmos de corrección de los errores asociados al proceso de captación y fusión



**Ilustración 28 : Esquema de ubicación de las cámaras para el modo *AumentoCampo***

Al igual que en el modo anterior, primero se hizo una aproximación burda para observar y analizar los principales problemas encontrados. La Ilustración 29, muestra un esquema de las operaciones realizadas.



**Ilustración 29 : Esquema inicial para el aumento de la resolución espacial**

En base a este primer esquema, se desarrolló una metodología para conseguir mejorar la secuencia final. Para ello es vital el proceso de registro de imágenes para la obtención de panorámicas, sobre el que existen infinidad de trabajos pero pocos lo abordan desde la perspectiva estudiada en este proyecto. La mayoría de trabajos consiguen una panorámica partiendo de las imágenes capturadas con una única cámara rotando sobre un eje imaginario que atraviesa la cámara. La región común entre dos



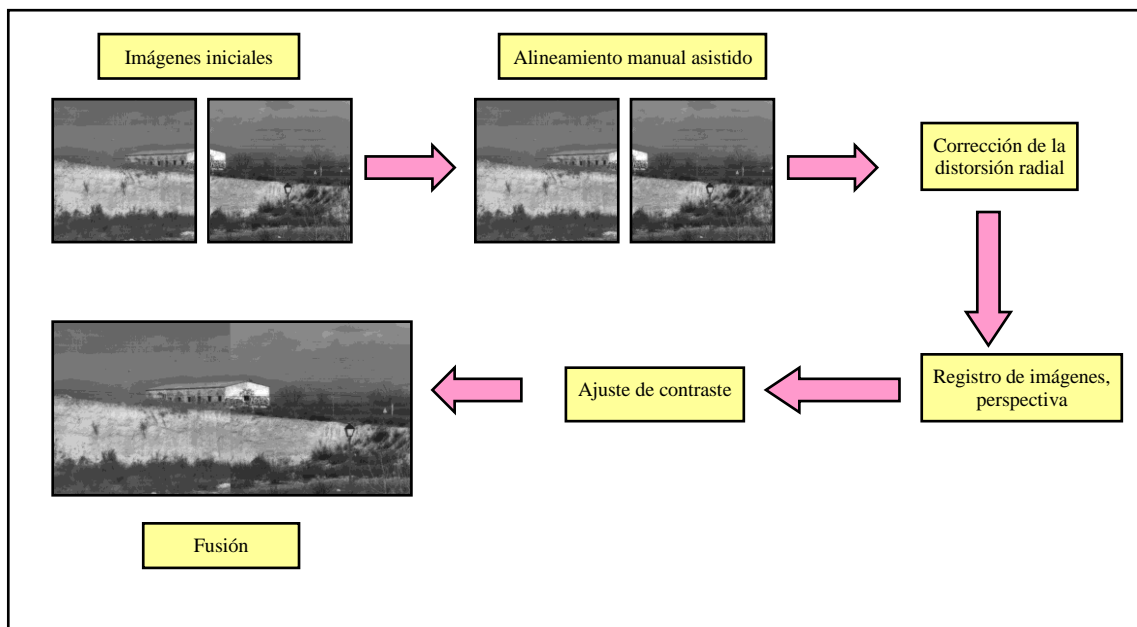
#### 4. Desarrollo de algoritmos de corrección de los errores asociados al proceso de captación y fusión

imágenes consecutivas representa más del 50 % de la imagen. Con estas condiciones existe mucha información redundante acerca de la escena para componer la imagen panorámica final. Ésta imagen es resultado de la fusión de imágenes capturadas en distintos tiempos, por lo que no es posible representar movimiento si se produce en la escena.

En este caso también es posible utilizar transformaciones del plano proyectivo de cada cámara para conseguir modificar el punto de vista de cada cámara y así corregir desviaciones debidas a la falta de paralelismo o alineación entre cámaras. Las correspondencias necesarias entre puntos se obtendrán de la región común entre dos imágenes de distintas cámaras contiguas, que en este caso representa el 10 % de la imagen. Hay que tener en cuenta que las zonas solapadas reducen la resolución de la imagen panorámica final pero que son necesarias para encontrar correspondencias y poder registrar las imágenes. De esta manera se consigue una imagen panorámica a partir de tantas imágenes individuales como cámaras conforman el sistema.

Además es especialmente necesario corregir la distorsión radial antes de registrar las imágenes para componer la panorámica, ya que esta distorsión es mas acentuada en los márgenes de la imagen , por la naturaleza de la distorsión radial,

Debido a la diferencia de configuración de los sistemas ópticos de cada cámara , aparecen diferencias de contraste entre las imágenes. Esta diferencia es apreciable en la región común de las imágenes, de donde se obtiene cuál es la diferencia de contraste entre las imágenes. Antes de realizar cualquier otra mejora, es necesario compensar esta diferencia de contraste entre las imágenes contiguas utilizando la información extraída de la region común. La Ilustración 30 muestra el esquema final implementado para obtener la secuencia final mejorada.



**Ilustración 30 : Diagrama de bloques para la obtención de una secuencia mejorada para el modo AumentoCampo**

#### 4.4. Algoritmo de alineamiento

Una primera aproximación al cálculo de la región de interés, necesario para el modo *AumentoTasa*, parte de la fundamentación teórica del proceso de captura de imágenes por una cámara descrita en el ANEXO A. En esta aproximación, las cámaras captan una parte de la escena. Según su ubicación en la estructura, la imagen común de cada cámara corresponde a una zona que se puede calcular a partir de la distancia focal de la lente utilizada ( $f$ ), la distancia a la escena ( $D$ ), la distancia entre las cámaras (Separación) y las dimensiones del sensor de la cámara ( $sensorDim$ ). Este algoritmo está implementado en la función, `\MultiMotion\CalcSep.m`

El cálculo que se realiza consiste en resolver el número de píxeles que se deben eliminar de cada imagen. Estos píxeles deberán ser eliminados de uno o dos de los lados de cada imagen dependiendo de la posición de la cámara en la estructura y del número de cámaras. Así bien para una configuración con 4 cámaras, las zonas a eliminar para cada posición en la cámara se muestran en la Ilustración 31.

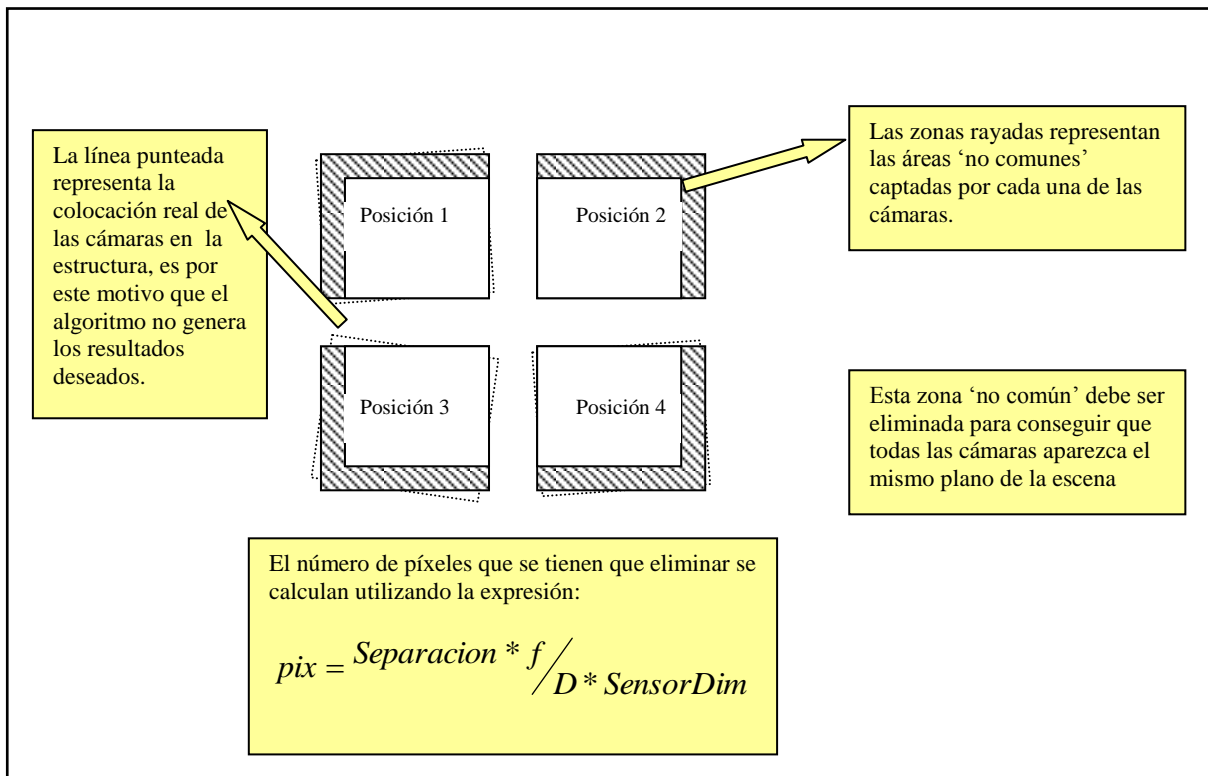


Ilustración 31 : Región común a imágenes capturadas en el modo *AumentoTasa*

Esta aproximación teórica al cálculo de la ROI, resulta poco adecuada ya que supone que las cámaras están perfectamente ubicadas dentro de la estructura, a nivel. En la realidad pequeñas desviaciones en la estructura hacen que las imágenes capturadas por cada cámara no estén totalmente alineadas y que el resultado de este algoritmo no genere los resultados deseados.

Para solucionar problemas derivados del no alineamiento de las cámaras dentro de la estructura, se optó por implementar un algoritmo que calculara la ROI común a las cuatro cámaras de una manera dinámica. El método consiste en identificar una región homogénea común a todas las imágenes y a continuación recortarlas para que la región

#### 4.Desarrollo de algoritmos de corrección de los errores asociados al proceso de captación y fusión

común tenga igual posición en todas ellas. Para ello se necesita conocer la posición exacta de dicha región en todas las imágenes. .

Así, en vez de tener que dar al algoritmo datos acerca de la escena que en ningún caso podemos conocer con exactitud, el usuario debe indicar una semilla (un punto cualquiera de la misma región en cada imagen) para que el algoritmo comience.

Esta semilla se introduce como entrada junto con la imagen en un algoritmo de crecimiento de regiones, implementado en la función \MultiMotion\growRegion.m. Este algoritmo, analiza los puntos vecinos al dado y comprueba si pertenecen o no a la región (para que pertenezca deben estar dentro de un rango de variación de luminancia, la media de los puntos de la región más un margen; se suele utilizar un 5 %).

Una vez obtenida la región, se calcula su centro de masas, Ilustración 32, punto que se asumirá como conocido con precisión en la imagen. Así se hará con el resto de capturas, partiendo de un punto homólogo que el usuario deberá indicar al algoritmo. De esta manera, se reduce el error a la hora de marcar los puntos homólogos en imágenes distintas.

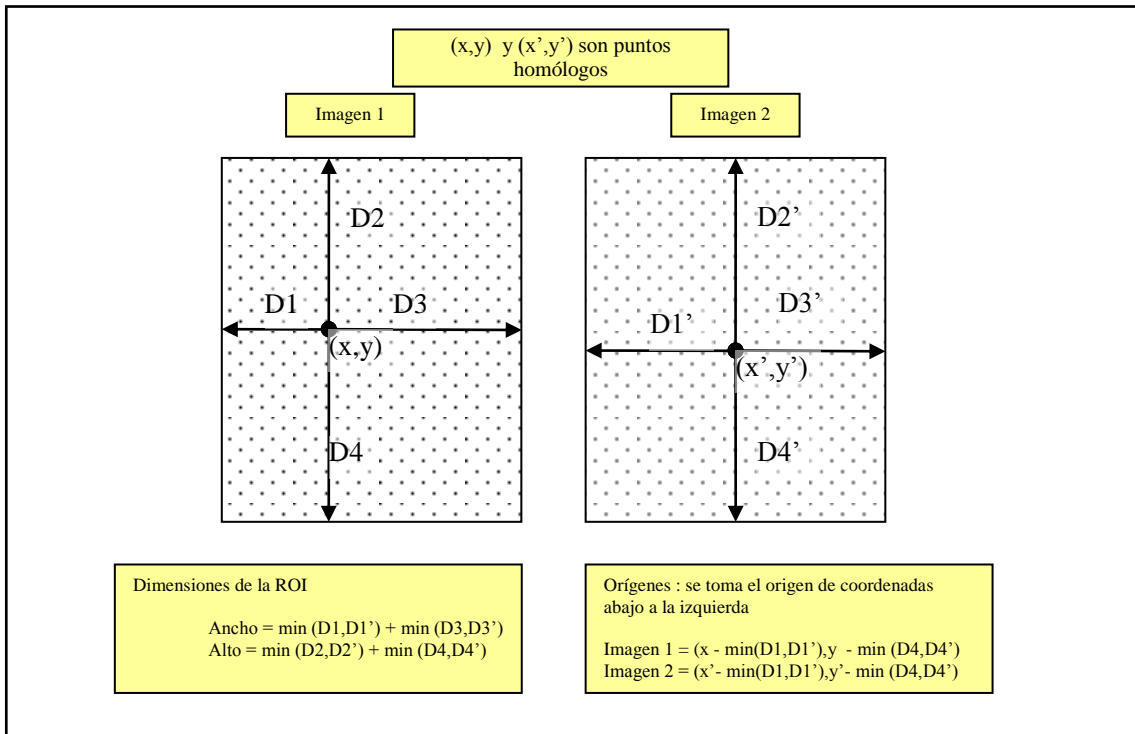
$$\vec{R}_{CM} = \frac{\sum_i (\vec{r}_i \cdot m_i)}{\sum_i m_i}$$

$m_i$  -> Masa puntual iésima  
 $\vec{r}_i$  -> Posición de la masa iésima respecto al eje de referencia asumido.

**Ilustración 32 : Formula utilizada para el cálculo del centro de masas**

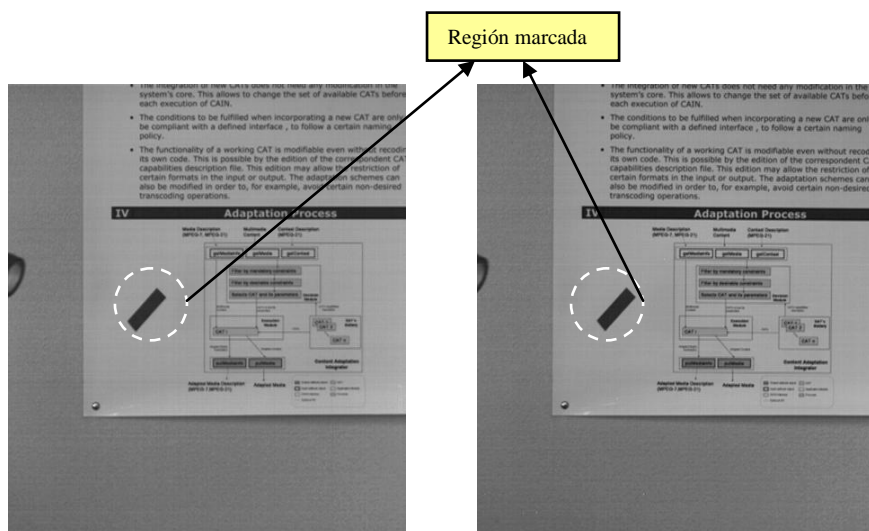
Una vez se tienen las coordenadas del punto homólogo en cada imagen unos pequeños cálculos generan los orígenes para cada imagen y las dimensiones de la región común, Ilustración 33.

#### 4. Desarrollo de algoritmos de corrección de los errores asociados al proceso de captación y fusión



**Ilustración 33 : Cálculo del origen y dimensiones de la región común a dos imágenes**

Estas operaciones se realizan en la función `calcOriginDimension` , donde se calcula el origen en cada una de las imágenes y la dimensión de la región común. En la Ilustración 34, se puede observar el resultado del algoritmo de alineamiento para dos imágenes capturadas por el sistema con dos cámaras.



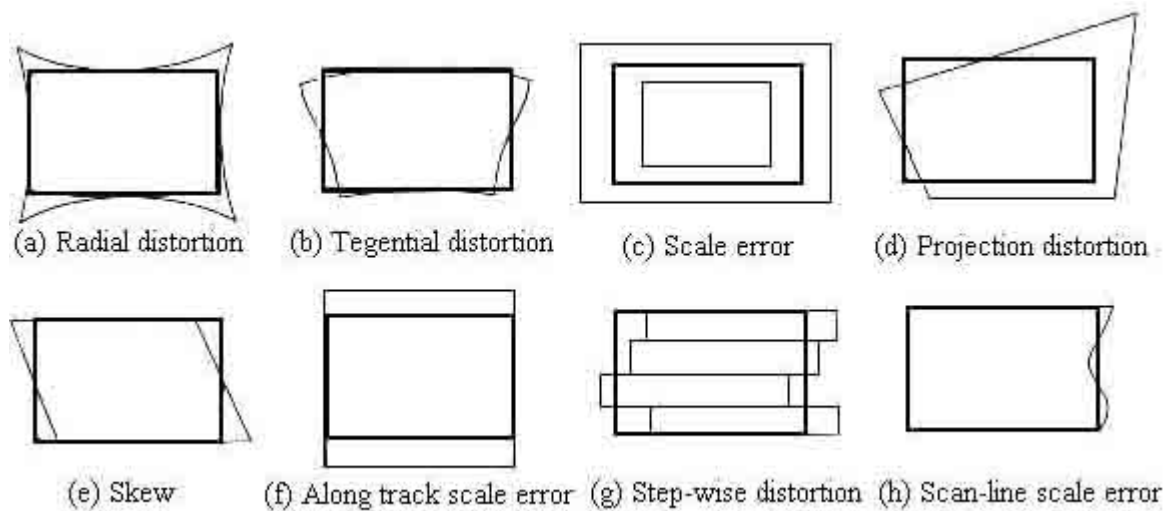
**Ilustración 34 : Resultado del algoritmo de alineamiento**

### 4.5. Corrección de la distorsión radial

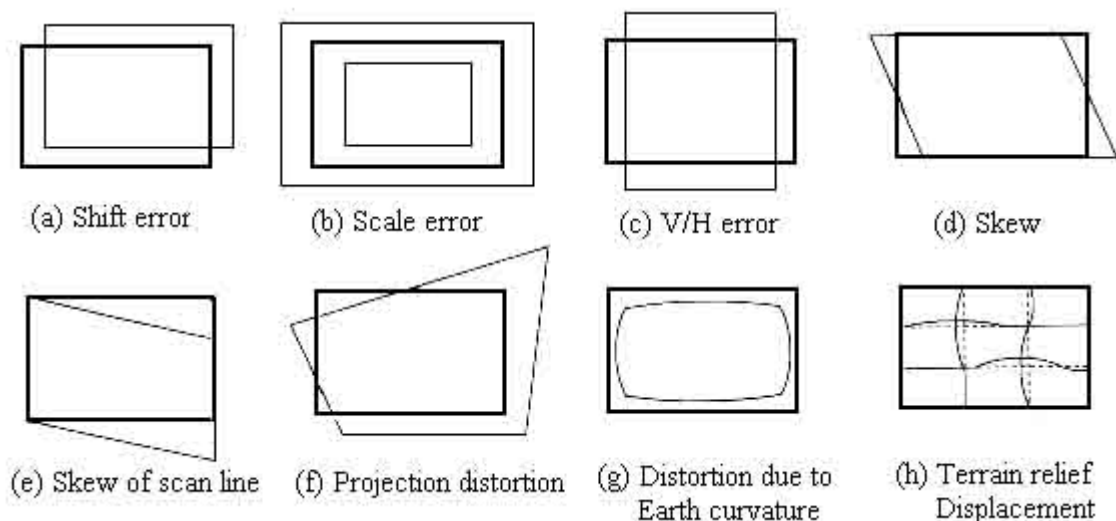
Una lente ideal usada en una cámara transforma líneas rectas como tal. Las lentes reales no son ideales e introducen modificaciones en la geometría de la imagen. La distorsión geométrica es un error en la imagen, entre las coordenadas de la imagen actual y las coordenadas de la imagen ideal. Hay diferentes tipos de distorsiones geométricas que están clasificadas en las siguientes categorías:

- Distorsión interna: resultado de la geometría de la lente, Ilustración 35
- Distorsión externa: resultado de la colocación de la lente y la forma del objeto. Ilustración 36

A continuación se muestran algunos tipos de distorsión interna y externa:



**Ilustración 35 : Algunos tipos de distorsión interna**



**Ilustración 36 : Algunos tipos de distorsión externa**

De todas las distorsiones no lineales presentadas, la distorsión radial es la mas importante y común.

#### 4.Desarrollo de algoritmos de corrección de los errores asociados al proceso de captación y fusión

Una cámara que presenta el fenómeno de distorsión radial, distorsiona las líneas rectas de la escena real, representándolas en sus imágenes como líneas curvas. Las líneas rectas que pasan por el centro de la imagen (que en la mayoría de los casos es el centro de la distorsión radial) permanecen rectas y una circunferencia concéntrica al centro de la imagen permanece con esta forma, pero modifica el valor del radio. Los típicos casos de distorsión radial son distorsión de tipo *barrel* y de tipo *pin-cushion*. Debido a que esta distorsión introduce no linealidades en la información geométrica de la escena real, es importante corregirla antes de abordar el problema de extraer información útil de la imagen o proceder a realizar cualquier otra mejora.

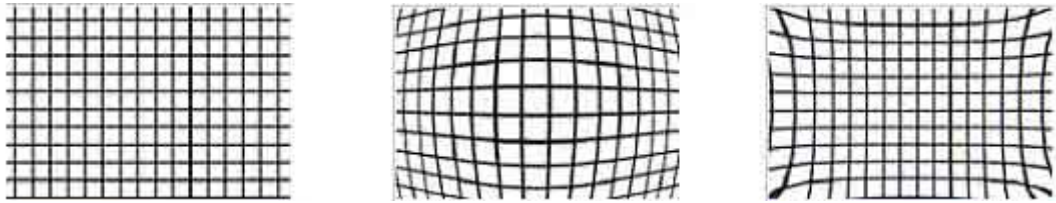


Ilustración 37 : Imágenes original, con distorsión *barrel* y *pin-cushion*

En este apartado, primero se hace una descripción del modelo matemático de la distorsión radial. En base a este modelo se presenta el método para determinar sus parámetros. Se describe la implementación de los algoritmos para la obtención del modelo de distorsión y de corrección de la imagen bajo el entorno de MATLAB.

##### 4.5.1. Modelo de distorsión radial

Para corregir la distorsión, se necesita primero modelarla. La mayoría de estudios consultados proponen que la distorsión radial sea modelada por una expresión polinómica de la forma:

$$r_u = f(r_d) = r_d + k_1 r_d^3 + k_2 r_d^5 + \dots, \text{ siendo} \quad \begin{array}{l} r_u = \text{radio no distorsionado} \\ r_d = \text{radio distorsionado} \end{array}$$

Esta expresión relaciona las distancias de los puntos distorsionados y corregidos respecto al centro de la distorsión. La distorsión es mínima cerca del centro de la imagen y aumenta al alejarse de este. Debido a la simetría radial de la distorsión, todos los coeficientes de orden par no aparecerán en la fórmula. En general, los trabajos publicados sobre la calibración de distorsión radial usan solamente dos coeficientes, pues con ellos se logra una buena aproximación; así de la fórmula anterior nos quedamos con:

$$r_u = r_d \left( 1 + k_1 r_d^2 + k_2 r_d^4 \right)$$

Los valores de los radios de los puntos corregidos (no distorsionados) y distorsionados se calculan

#### 4.Desarrollo de algoritmos de corrección de los errores asociados al proceso de captación y fusión

$$r_d = \sqrt{(x_d - x_c)^2 + (y_d - y_c)^2} \quad r_u = \sqrt{(x_u - x_c)^2 + (y_u - y_c)^2}$$

, siendo:

- $x_c, y_c$  el centro de la distorsión radial, que para un caso inicial tomaremos como el centro de la imagen o punto principal.
- $p_d = (x_d, y_d)$  las coordenadas del punto distorsionado.
- $p_u = (x_u, y_u)$  las coordenadas del punto corregido o no distorsionado.

#### 4.5.2. Obtención de los parámetros de la distorsión radial

Existen trabajos como el de Zhang [8] el cual propone un método de calibración y modelado de cámara, donde también se modela la distorsión radial. Sin embargo, este requiere el conocimiento de algunas características físicas que al menos en nuestro caso no siempre estarán disponibles. Por esta razón, se necesita un modelo de distorsión y métodos de corrección de imágenes los cuales no requieran de información a priori de la cámara, sino que obtenga la información solamente de la imagen distorsionada.

El método que aquí se presenta utiliza información *a priori* de la escena real. Aprovecha características geométricas de los objetos capturados en la imagen, linealidad y simetría. Para este estudio, se utiliza un damero, compuesto por cuadrados y se aprovecha precisamente el alineamiento de los cuadrados, Ilustración 38



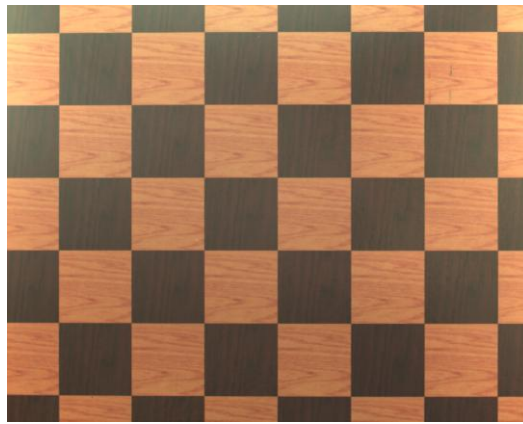
**Ilustración 38 : Estructura utilizada para el cálculo de los parámetros de distorsión radial**

Para la obtención de los parámetros que modelan la distorsión radial, se parte de una imagen capturada por la cámara de la estructura utilizada, el damero. Esta imagen



#### 4.Desarrollo de algoritmos de corrección de los errores asociados al proceso de captación y fusión

únicamente debe representar una parte de la estructura utilizada, tal y como muestra la Ilustración 39 .A continuación se extraen las coordenadas de los puntos característicos, esquinas.



**Ilustración 39: Ejemplo de imagen capturada para el algoritmo de corrección radial**

Estos puntos, en la realidad, están alineados debido a la forma del patrón o estructura utilizada. Este método busca la combinación de los parámetros que modelan la distorsión que hacen que la linealidad de estos puntos mejore en la imagen del patrón. Para ello, se define una función que determina el error de linealidad de estos puntos y a partir de la misma se trabaja para reducir este error hasta el mínimo.

Una vez obtenidos los parámetros que modelan la distorsión radial, validos únicamente para el modelo y ópticas utilizadas, pueden ser utilizados para corregir cualquier imagen capturada por el sistema cámara-óptica.

#### **4.5.3. Implementación del algoritmo de modelado y corrección de la distorsión radial**

El algoritmo desarrollado consta de diferentes etapas, que se describen continuación:

- Construcción del patrón y captura de su imagen.
- Localización de puntos característicos.
- Estimación de los coeficientes de la distorsión radial.
- Corrección de la distorsión radial.

*Construcción del patrón.* El patrón utilizado debe contener líneas que sean paralelas o perpendiculares entre sí. La distancia que separan las líneas paralelas debe ser la misma de modo que el patrón sea similar una cuadrícula. De esta manera la extracción de los puntos de corte de las líneas resulta más sencilla. Estos puntos se utilizarán para definir las rectas representadas en la imagen y calcular el grado de linealidad.

Una primera estructura utilizada , una hoja DIN A4 con cuadrados negros y blancos impresos en él, no aseguraba la linealidad del patrón ya que en la colocación del papel



#### 4.Desarrollo de algoritmos de corrección de los errores asociados al proceso de captación y fusión

no se conseguía una estructura totalmente plana y no se mantenían las propiedades deseadas.

La utilización de un damero rígido asegura que las líneas representadas en la estructura pertenezcan todas al mismo plano y al ser representadas en una imagen aparecerán como líneas rectas según la geometría proyectiva (en realidad esto no ocurre así debido a la distorsión).

El patrón utilizado será una imagen del damero utilizado, paralelo en la medida de lo posible al sensor de la cámara y con una orientación nivelada. La iluminación del damero debe ser lo mas homogénea posible para evitar introducir ruido en la imagen que luego provoque errores en la etapa de detección de puntos característicos.

##### *Localización de puntos característicos.*

Esta etapa consiste en encontrar las coordenadas de las esquinas del damero (u otro patrón similar) representado en la imagen. Existen muchos métodos que detectan líneas y puntos característicos como Sobel, Prewitt, Roberts, Canny...Sin embargo, en este caso, puesto que tenemos control sobre parámetros como la iluminación y conocemos a priori la ubicación de las esquinas en la imagen, resulta muy sencillo localizar las esquinas del damero si filtramos la imagen con un filtro o máscara del tipo:

$$filtro = \begin{pmatrix} 1 & 1 & 0 & -1 & -1 \\ 1 & 1 & 0 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 1 & 1 \\ -1 & -1 & 0 & 1 & 1 \end{pmatrix}$$

En los bordes, se utiliza un replicado de la imagen para evitar que detecte como esquinas puntos que pertenecen al borde. MATLAB tiene implementada una función para realizar esta operación

`Imfiltrada=imfilter(im,filtro,'replicate')`

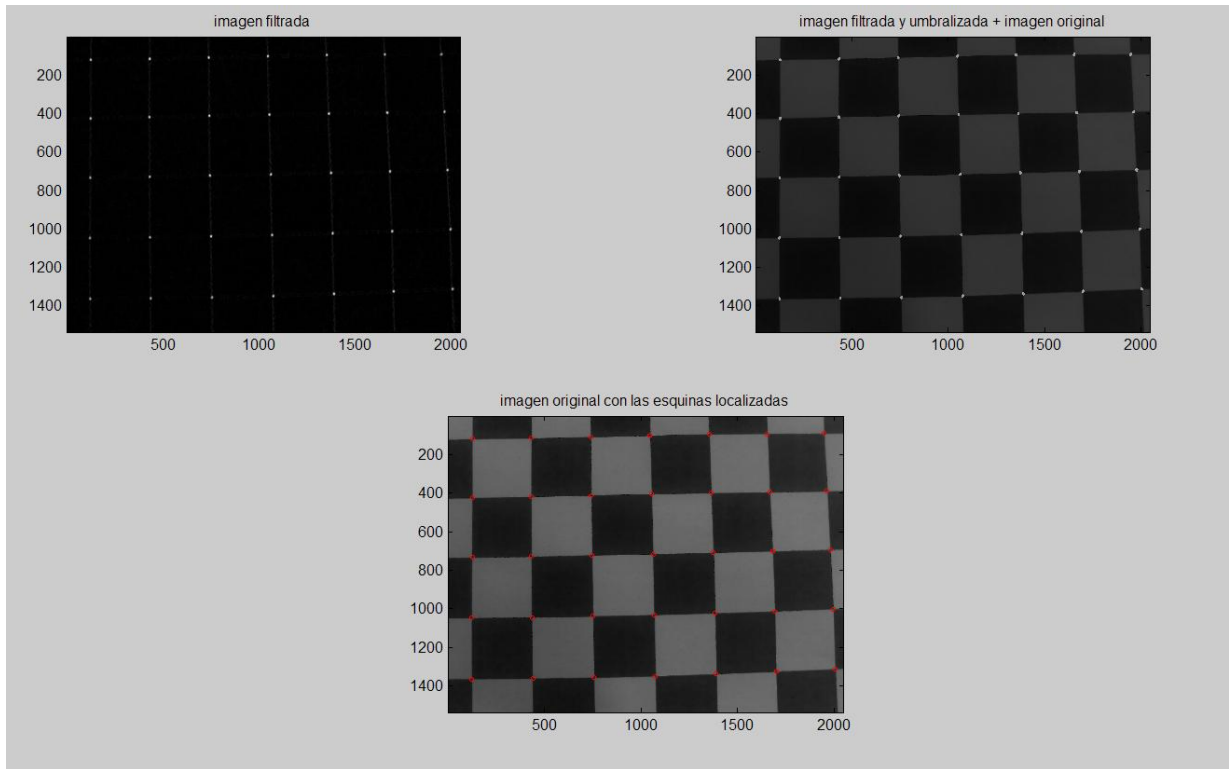
Una vez realizado el filtrado (Ilustración 40), se normaliza la imagen de manera que el valor máximo de intensidad en la imagen sea 255 y el mínimo 0. En esta imagen, los máximos coinciden con las esquinas del patrón, los mínimos serán zonas homogéneas de la imagen. A continuación elegimos un umbral (20 % del máximo) y pasamos la imagen a binario (0 o 1) utilizando este umbral como límite.

Sobre la imagen umbralizada buscamos las coordenadas del primer máximo (los máximos son los píxeles con valor 1, en teoría tan solo se encuentran agrupados en las proximidades de las esquinas). Introduciendo éstas coordenadas como semilla y la imagen umbralizada invocamos al algoritmo de crecimiento de regiones

#### 4.Desarrollo de algoritmos de corrección de los errores asociados al proceso de captación y fusión

(*growRegion.m*) para estimar la región y calcular su centro de masas , que serán las coordenadas del punto de cruce

A continuación, eliminamos ésta región de la imagen umbralizada, buscamos de nuevo el primer máximo e invocamos de nuevo al algoritmo de crecimiento de regiones. Así hasta que tenemos detectados todos los puntos característicos de la imagen, es decir tenemos todos los puntos de cruce.



**Ilustración 40 : Etapa de detección de puntos característicos: imagen filtrada (arriba izda.), imagen filtrada y umbralizada junto con la imagen original (arriba dcha.), e imagen original con los puntos característicos localizados (abajo)**

Una vez tenemos estas coordenadas, antes de pasar a la siguiente etapa deben ser colocados en una matriz. Las dimensiones de esta matriz serán  $M \times N \times 2$  donde  $M$  es el número de puntos verticales (líneas horizontales) que se visualizan en la imagen (5 en el ejemplo, Ilustración 40) y  $N$  el número de punto horizontales (líneas verticales que contiene la imagen . 7 en el ejemplo, Ilustración 40). La primera dimensión de esta matriz de  $M \times N$  contiene los valores de la coordenadas de las  $x$  y la segunda las de las  $y$ . El número de líneas horizontales y verticales,  $M$  y  $N$ , deben ser indicados por el usuario.

#### *Estimación de los coeficientes de la distorsión radial*

Una vez conseguidas las coordenadas sobre la imagen de los puntos característicos que forman el patrón o plantilla y considerando únicamente como información a priori que estos puntos han de estar alineados, se describe a continuación un algoritmo que realiza una estimación de los coeficientes que modelan la distorsión radial.

#### 4. Desarrollo de algoritmos de corrección de los errores asociados al proceso de captación y fusión

La idea consiste en minimizar el error de linealidad. Este error se define como la suma de las distancias de los puntos a la recta; ésta se trata de una aproximación por regresión lineal de los puntos que la componen. Puesto que la linealidad debe permanecer en ambas dimensiones, horizontal y vertical, estos cálculos se realizan en ambas.

$$err_l = \sum_{i=1}^{numLinH} \sum_{j=1}^{numPoinV} d(x_{ij}, lineH_i) + \sum_{i=1}^{numLinV} \sum_{j=1}^{numPoinH} d(x_{ij}, lineV_i)$$

, donde:

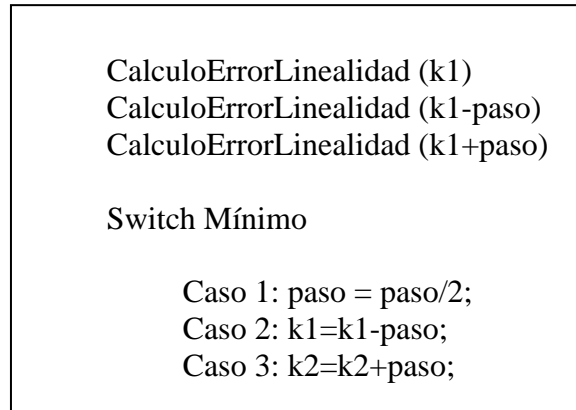
- $d(x_{ij}, lineH_i)$  es la distancia del punto  $x_{ij}$  a la recta representada por  $lineH_i$ ; ésta recta se obtiene haciendo una estimación por regresión lineal de los puntos que la componen (puntos de corte calculados), una fila o columna completa en la matriz de datos .
- $numLinH$  indica el número de líneas horizontales que se visualizan en la plantilla que coincide con el número de punto verticales,  $numPoinV$ , que hay detectados en la imagen.
- $numLinV$  indica el número de líneas verticales que se visualizan en la plantilla que coincide con el número de punto horizontales,  $numPoinH$ , que hay detectados en la imagen.

Lo primero es realizar una estimación de los coeficientes, lo suficientemente buena para que el método de búsqueda *simplex*, que localiza mínimos locales de una función dada a partir de unos valores iniciales, genere resultados coherentes.

Para encontrar una buena aproximación para el primer coeficiente, se implementó una adaptación del algoritmo de descenso de gradiente. Éste método consiste en ir calculando el error de linealidad para 3 valores del coeficiente, quedarse con el menor y volver a iterar. Para esta aproximación, el valor del segundo coeficiente se establece a 0.

Los valores que irá tomando el primer coeficiente, dependerán del resultado de la iteración anterior. Primero, se elije un paso, normalmente del orden de  $10^{-3}$ , aunque puede variar para distintos casos. También ha de elegirse un valor inicial para el primer coeficiente,  $k_1 = 0$ . La Ilustración 41 muestra las operaciones en una iteración .

#### 4.Desarrollo de algoritmos de corrección de los errores asociados al proceso de captación y fusión



**Ilustración 41 : Iteración para la estimación del primer coeficiente, k1, que modela la distorsión radial**

Con este método, los resultados convergen hacia un valor en unas 25 iteraciones. Con la estimación del primer coeficiente y 0 para el segundo coeficiente, se introduce como valores iniciales en el método de búsqueda simplex, que encuentra el mínimo local de la función entorno a ese valor inicial.

Matlab tiene una colección de funciones (fminsearch, fminbnd, fmincon, fminunc) para calcular el mínimo de funciones con múltiples variables. El método utilizado para esta búsqueda (Newton , simplex, PCG , SQP ....) puede ser escogido por el usuario pero Matlab elije por defecto el más óptimo para el problema planteado, condiciones y variables iniciales.

#### *Corrección de la distorsión radial*

En esta etapa, utilizando los coeficiente obtenidos en la etapa anterior , se describe cómo corregir la distorsión de cualquier imagen capturada por la cámara-lente. En teoría estos resultados son únicos para cada cámara-lente , aunque resultaron ser muy similares para distintas camaras utilizadas con la misma lente. Este proceso resulta caro en términos de tiempo de ejecución puesto que es necesario realizar primero un cálculo para conocer donde se desplaza cada punto y luego es necesario realizar una interpolación de dos dimensiones.

En el primer paso, se necesita resolver la siguiente ecuación para todas las coordenadas de la imagen corregida:

$$(x_u - x_c, y_u - y_c) = (x_d - x_c, y_d - y_c) * (1 + k_1 r_d^2 + k_2 r_d^4)$$

, donde:

- $X_u = (x_u, y_u)$  son las coordenadas del puntos corregido
- $X_d = (x_d, y_d)$  son las coordenadas del punto distorsionado
- $X_c = (x_c, y_c)$  son las coordenadas del centro óptico; en este trabajo se trata del centro de la imagen, por simplificación.
- $r_d$  es el radio del punto distorsionado

#### 4.Desarrollo de algoritmos de corrección de los errores asociados al proceso de captación y fusión

Con esta fórmula, se calcula cuáles son las coordenadas de la imagen original (por ejemplo [1.25,1.25] ), que corresponde con la coordenada entera de la imagen corregida ( por ejemplo [1,1]). Es necesario realizar cinco o más iteraciones del método de Newton para encontrar los valores de  $(x_d, y_d)$  que cumplen la ecuación anterior con un error mínimo. Para ciertos casos, donde la distorsión introducida por las lentes resulta casi inapreciable, es posible aproximar esta ecuación a:

$$(x_d - x_c, y_d - y_c) = (x_u - x_c, y_u - y_c) * (1 - k_1 r_u^2 - k_2 r_u^4)$$

, siendo  $r_u$  el radio del punto corregido. Con esta aproximación, el valor de la coordenada no distorsionada se obtiene directamente sin necesidad de realizar una búsqueda del valor que mas se aproxima. Con una imagen de 1024 x 1024 píxeles de resolución, hay que realizar esta operación aproximadamente un millón de veces. Teniendo en cuenta que la aproximación consume seis veces menos de tiempo que la búsqueda exacta y que calcular el resultado de formular tarda aproximadamente 10 ms. El tiempo necesitado para calcular la matriz de corrección es 2'5 y 0'4 horas respectivamente.

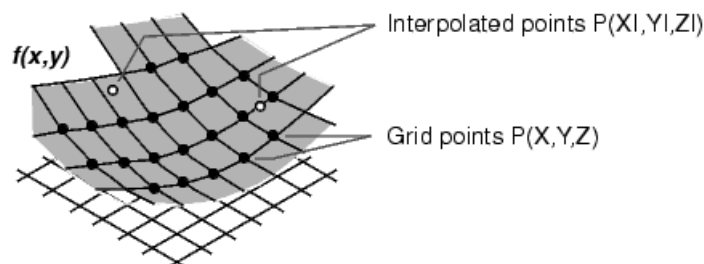
Como resultado de aplicar la formula en todas las coordenadas de la imagen, se obtiene una matriz de correspondencia entre los valores de la imagen no distorsionada y distorsionada. A continuación se muestra un ejemplo de la matriz.

$$\text{Correction (1 , 1 , :) = (1.8849, 1.1662)}$$

El valor de la coordenada (1,1) de la imagen no distorsionada, corregida, corresponde con el valor de la coordenada (1.8849, 1.1662) de la imagen distorsionada.

Una vez obtenida esta matriz, una interpolación en dos dimensiones a partir de la imagen distorsionada que deseamos corregir nos dará los valores de la imagen corregida. El método de interpolación puede ser desde el más sencillo, el vecino más próximo, u otros más complejos, como una interpolación bicúbica.

Para este trabajo ha sido elegida la interpolación bilineal, que siendo una de las más económicas en términos computacionales, produce resultados poco mejorables por métodos más complejos. Se utiliza una función de MATLAB que realiza esta operación de manera óptima, Ilustración 42.



**Ilustración 42 : Comando en Matlab que realiza una interpolación de segundo orden, interp2(Z,XI,YI)**

#### 4.5.4. Resultados, conclusiones y trabajo futuro

Este apartado sobre la corrección radial se puede dividir en tres etapas. En la primera, se obtiene puntos característicos de la estructura utilizada, una segunda donde se calculan los parámetros que modelan la distorsión radial. Y una tercera etapa donde, una vez conocidos los parámetros obtenidos en la fase anterior, se corrige cualquier imagen generada por el sistema cámara-lente utilizado en la primera etapa (ver Ilustración 43).

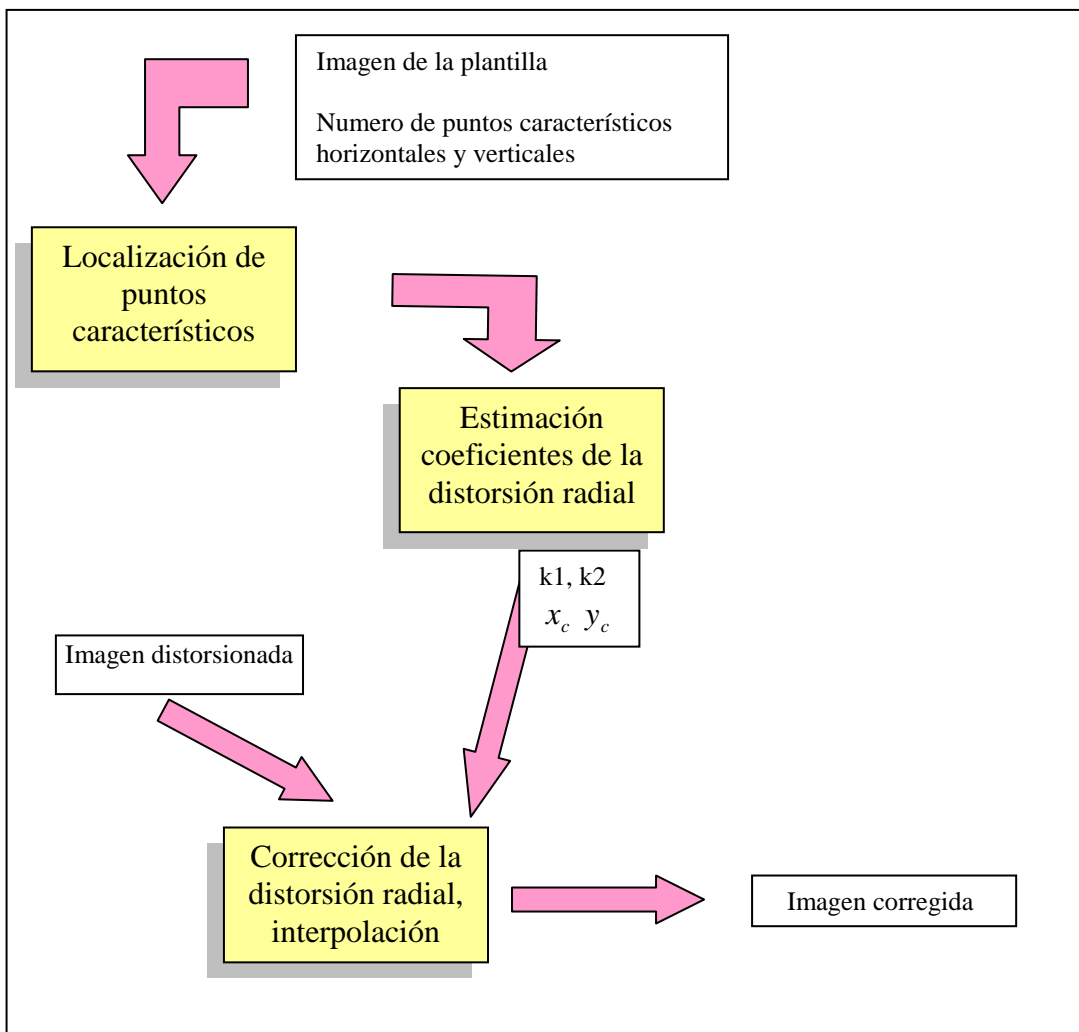
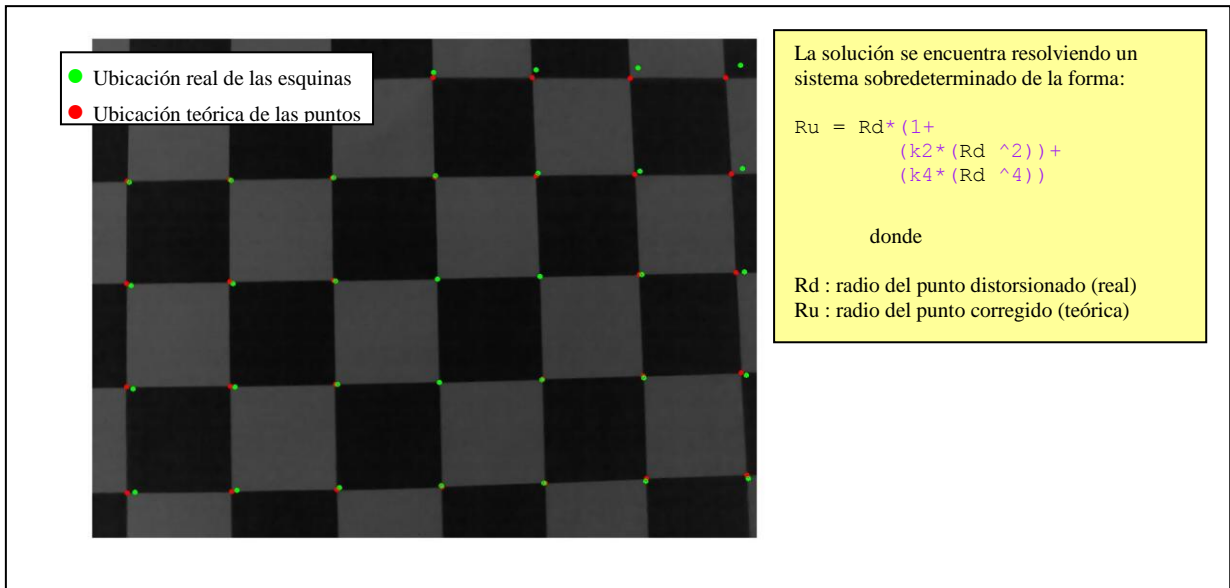


Ilustración 43 : Resumen del algoritmo que modela y corrige la distorsión radial

#### *Aproximación inicial*

En una primera aproximación a la estimación de los coeficientes de la distorsión radial expuestos en este trabajo, e intentando aprovechar las propiedades de ortogonalidad de la estructura utilizada, un damero, se realizaba una estimación de la ubicación teórica de los puntos característicos. A continuación se resolvía el sistema de ecuaciones planteado y se obtenían los coeficientes de la distorsión, Ilustración 44.

#### 4.Desarrollo de algoritmos de corrección de los errores asociados al proceso de captación y fusión



**Ilustración 44: Aproximación inicial a la estimación de los coeficientes de la distorsión radial.**

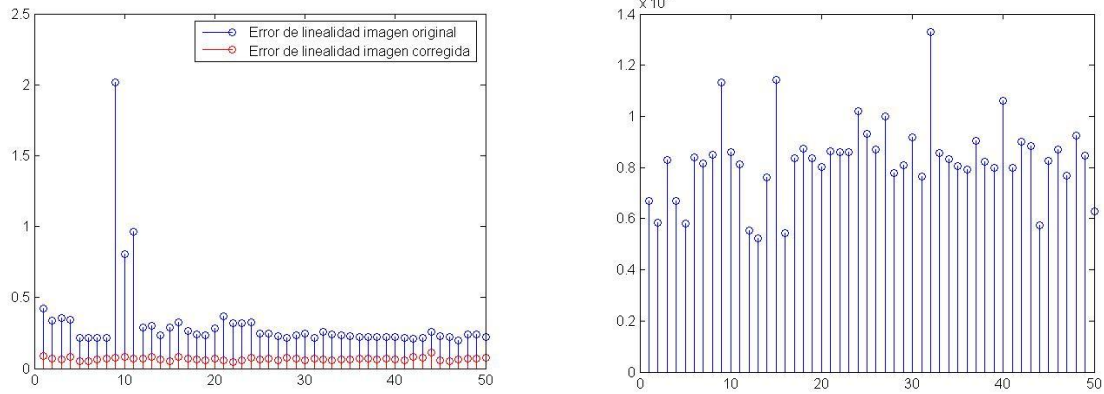
La estimación de las coordenadas teóricas resultó ser más complicada de lo que en un principio parecía. Para que este cálculo resulte bastante aproximado a la realidad, hay que tener en cuenta muchos parámetros relacionados con la ubicación de la estructura con respecto al plano del sensor de la cámara en el momento de la captura, dimensiones de la estructura, ángulos, origen de referencia...

#### *Pruebas sobre la solución adoptada*

Para validar el algoritmo desarrollado en este apartado, ha sido desarrollado en varios *script* y funciones en Matlab. Estos archivos se encuentran ubicados en la carpeta /DistorsionRadial que se adjunta junto con este documento. Para probar, ejecutar *test.m* desde la consola de Matlab. Al comienzo de este *script* se encuentran las variables necesarias para que sea ejecutado correctamente. Los resultados para algunas pruebas realizadas utilizando estos archivos se exponen a continuación.

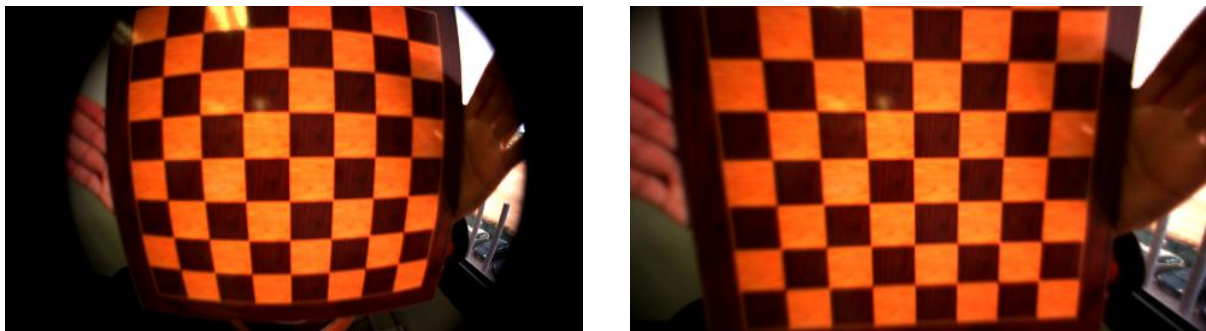
Para la verificar que la segunda etapa generaba resultados coherentes se realizaron diversas pruebas. Una primera consistía en observar la estabilidad del método para distintas imágenes de la plantilla captadas con el mismo sistema cámara-lente. Las siguientes figuras muestran los resultados de pruebas realizadas a 50 imágenes capturadas con el mismo sistema cámara-lente, 27 mm, con distinta orientación y enfoque.

#### 4. Desarrollo de algoritmos de corrección de los errores asociados al proceso de captación y fusión



**Ilustración 45: Resultado del algoritmo para 50 imágenes: mejora del error de linealidad (izda.), y valores obtenidos para el primer coeficiente (dcha.).**

Puesto que las lentes utilizadas en el sistema, no introducían una distorsión realmente apreciable, se tuvo que comprobar que el algoritmo era capaz de corregir imágenes con una distorsión muy acusada. Para ello se capturó el damero utilizado con una lente de 4.5 mm y se aplicó el algoritmo, produciendo los siguientes resultados.



**Ilustración 46: Resultados para una lente de 4.5 mm: imagen capturada (izda.) e imagen corregida (dcha.).**

La Tabla 10 , muestra el tiempo de ejecución aproximado empleado para realizar cada una de las etapas. Estas mediciones se realizaron utilizando 30 puntos característicos (6 columnas por 5 filas) y unas 50 imágenes (resolución de 1280 x 1024 píxeles). Las 3 primeras columnas indican el tiempo que tarda el algoritmo en calcular los parámetros de la distorsión y la matriz de correspondencia. La última columna indica el tiempo que se consume para corregir cualquier imagen.

**Tabla 10 : Tiempo de ejecución del algoritmo de corrección radial**

<i>Tiempo de ejecución, aprox \ Etapa</i>	<i>Localización de puntos característicos</i>	<i>Estimación de los coeficientes</i>	<i>Calculo de la matriz de corrección método aproximado / método exacto</i>
<i>Segundos (s.)</i>	<i>13</i>	<i>7</i>	<i>37 /3000</i>



#### 4.Desarrollo de algoritmos de corrección de los errores asociados al proceso de captación y fusión

##### *Conclusiones*

Las variables que modelan la distorsión radial, según esta expuesto al principio de este apartado, son los dos primeros coeficientes,  $k_1$  y  $k_2$ , y el centro de la distorsión.

Desde un principio, el centro de la distorsión ha sido considerado el centro de la imagen. Aunque esto no es del todo correcto, es una muy buena aproximación teniendo en cuenta que existen otros parámetros más determinantes ( $k_1$  y  $k_2$ ) y que la calidad de las cámaras y lentes con que se trabaja es elevada.

Después de analizar los resultados de la pruebas, se optó por no utilizar el segundo coeficiente. En los casos analizados, donde las ópticas utilizadas (27, 50 y 105 mm) no introducen grandes distorsiones, la utilización de un único coeficiente para modelar la distorsión resulta suficiente. La utilización del segundo coeficiente no produce mejoras significativas.

En la práctica, para corregir la distorsión es necesario interpolar para conocer los valores de las nuevas coordenadas. Esta interpolación se realiza utilizando lo que en este documento se ha denominado matriz de correspondencia, que relaciona los puntos de la imagen corregida con puntos de la imagen original.

Con las lentes utilizadas, el error de linealidad inicial para las imágenes capturadas por las cámaras resulta aproximadamente de 0.5 píxeles. Este error, utilizando únicamente el primer coeficiente se reduce a valores menores de 0.1 píxeles. Este pequeño error residual, es debido a otro tipo de distorsiones introducidas por la lente que este trabajo no aborda, a las citadas suposiciones acerca del centro de la distorsión y a la no utilización del segundo coeficiente (y otros de mayor orden).

##### *Trabajo Futuro*

Siguiendo la línea comenzada en este trabajo, sería posible mejorar el algoritmo en las tres etapas: en la primera, aumentar la autonomía y evitar la necesidad de indicar al algoritmo el número de líneas verticales y horizontales que se visualiza en la imagen. En la segunda etapa, eliminando la obligación de utilizar una estructura matricial o patrón con los puntos conocidos. Quizás sería interesante analizar la capacidad del algoritmo para generar resultados utilizando tan solo 5 ó 6 puntos pertenecientes a una única recta que formara parte de la escena, sin necesidad de utilizar un patrón determinado. Ponderar la utilidad del segundo coeficiente ( $k_2$ ) y otros de mayor orden, para aquellas ópticas donde la distorsión sea más acentuada. También, evaluar la utilidad del valor real del centro de la distorsión; este valor debería ser obtenido siguiendo la línea seguida en este trabajo acerca de la distorsión. Es decir, sin el conocimiento previo de parámetros ni características de la escena de la imagen sobre la que se trabaja

#### **4.6. Ajuste de perspectiva**

#### 4.Desarrollo de algoritmos de corrección de los errores asociados al proceso de captación y fusión

En este apartado se introduce las principales ideas sobre geometría proyectiva, en particular la geometría de las transformaciones proyectivas del plano. Bajo una cámara de perspectiva, algunas propiedades geométricas se conservan, tales como colinealidad (una línea recta se proyecta en una recta), mientras otras propiedades no, por ejemplo paralelismo, en general las líneas paralelas no se presentan como tales en la imagen. La geometría proyectiva modela el proceso de adquisición de la imagen al mismo tiempo que da una representación matemática apropiada para los cálculos.

Podemos utilizar las herramientas desarrolladas en este apartado para conseguir un ‘alineamiento’ mas preciso (unificación del punto de vista) de las cámaras implicadas en el modo *AumentoTasa*. También estas herramientas pueden ser utilizadas en el modo *AumentoCampo*, para alinear la perspectiva de las cámaras, es decir simular que la orientación de la camaras respecto al plano donde se desarrolla la escena es ideal, A continuación se fusionan las imágenes y se compone la panorámica para cada instante de tiempo. La obtención de resultados depende fuertemente de las variables iniciales del problema, es decir, de la obtención de puntos correspondientes entre las imágenes de cada cámara.

##### 4.6.1. La transformación del plano proyectivo

Una transformación proyectiva del plano proyectivo  $P^2$  transforma un plano en otro plano equivalente en el que se conservan todas las propiedades invariantes a las proyectividades. Dados un conjunto de puntos  $x_i$  en  $P^2$  y otro de puntos correspondientes  $x'_i$  también en  $P^2$  se desea la transformación proyectiva que toma cada correspondencia  $\{ x_i \leftrightarrow x'_i \}$ . En una situación práctica, los puntos  $x_i, x'_i$  representan puntos en dos imágenes, cada una de una cámara. Cada imagen es considerada un plano proyectivo  $P^2$  de una misma escena. El problema consiste en calcular la matriz  $H$ , con dimensiones  $3 \times 3$ , que cumple  $x'_i = H x_i$  para cada  $i$ .

La primera cuestión a considerar es cuantas correspondencias  $\{ x_i \leftrightarrow x'_i \}$  son necesarias para calcular la transformación proyectiva  $H$ . Por un lado, la matriz  $H$  tiene 9 valores, pero se mantiene invariante ante una multiplicación de todos sus elementos por una constante. Entonces el número de grados de libertad de una transformación proyectiva en 2D es 8. Por otro lado, cada punto  $x_i$  tiene dos componentes que pueden ser computadas por separado. Entonces, son necesarias 4 correspondencias  $\{ x_i \leftrightarrow x'_i \}$  para completar la matriz  $H$ .

Si se tienen 4 correspondencias, es posible encontrar una solución para la matriz  $H$ . Esta sería la solución mínima. Esta solución es valida si se tiene un algoritmo robusto que obtenga correspondencia entre puntos exacta. Sin embargo, siempre esta medición resulta inexacta (debido a ruido de diversa procedencia) y es necesario utilizar un numero mayor de correspondencias  $\{ x_i \leftrightarrow x'_i \}$ . Ahora, tenemos un sistema sobredeterminado y hay que encontrar la ‘mejor’ aproximación con los datos iniciales. Normalmente esto se realiza buscando una transformación  $H$  que minimiza una función de coste.

El algoritmo de Transformación Lineal Directa (DLT) [4] es un algoritmo lineal sencillo para determinar  $H$  utilizando correspondencias entre puntos. Antes de ejecutar

#### 4.Desarrollo de algoritmos de corrección de los errores asociados al proceso de captación y fusión

el algoritmo, que será descrito a continuación, es necesario realizar una normalización de los datos iniciales. Una vez obtenida la matriz  $\mathbf{H}$ , hay que desnormalizarla utilizando la operación inversa a la normalización realizada inicialmente. Esta normalización y desnormalización de los datos iniciales resulta imprescindible para obtener resultados.

### 4.6.2. DLT normalizado para el cálculo de homografías 2D

Dados  $n > 4$  correspondencias de puntos 2D,  $\{x_i \leftrightarrow x'_i\}$ , buscamos una matriz  $\mathbf{H}$ , denominada homografía, que cumpla que  $x'_i = Hx_i$ , para todas las parejas de puntos. El algoritmo que resuelve este problema esta dividido en las siguientes etapas:

**Normalización de  $x_i, x'_i$ :** Calcula una transformación equivalente  $\mathbf{T}, \mathbf{T}'$  (una para cada conjunto de puntos) que traslada y escala la representación de los citados conjuntos de puntos  $x_i, x'_i$ . Esta operación transforma las coordenadas de los puntos  $x_i, x'_i$  a un nuevo conjunto de puntos  $\tilde{x}_i, \tilde{x}'_i$ , donde el centroide de los puntos  $\tilde{x}_i, \tilde{x}'_i$  es la coordenada en el origen  $(0,0)^T$ , y la distancia media al origen sea  $\sqrt{2}$ .

$$x_i \mathbf{T} = \tilde{x}_i \quad \text{y} \quad x'_i \mathbf{T}' = \tilde{x}'_i$$

$x_i, x'_i, \tilde{x}_i, \tilde{x}'_i$  son representados como  $(x, y)$

**Calculo de la matriz  $\mathbf{A}_i$ :** para cada correspondencia  $\{\tilde{x}_i \leftrightarrow \tilde{x}'_i\}$ , calcula la matriz  $\mathbf{A}_i$ . Aunque se trata de un sistema con tres ecuaciones, únicamente dos de ellas son linealmente independientes. Por lo tanto se omitirá la tercera ecuación planteada del sistema:

$$\mathbf{A}_i \mathbf{h} = 0$$

Donde

$$\mathbf{A}_i = \begin{bmatrix} 0^T & -w'_i X_i^T & y'_i X_i^T \\ w'_i X_i^T & 0^T & -x'_i X_i^T \\ -y'_i X_i^T & x'_i X_i^T & 0^T \end{bmatrix}$$

En esta ecuación han sido omitidos los corchetes,  $\sim$ , que indican que se tratan de coordenadas normalizadas para evitar confusiones.

Los puntos utilizados  $\{\tilde{x}_i \leftrightarrow \tilde{x}'_i\}$  y  $\mathbf{0}$ , están representados en la fórmula en un sistema de coordenadas homogéneo de la forma  $\mathbf{X}_i^T = (x_i, y_i, w_i)$ ,  $\mathbf{X}'_i^T = (x'_i, y'_i, w'_i)$  y  $\mathbf{0}^T = (0,0,0)$  respectivamente. Se puede elegir  $w_i = 1$ , que significa que  $(x_i, y_i)$  son las coordenadas medidas sobre la imagen.

#### 4.Desarrollo de algoritmos de corrección de los errores asociados al proceso de captación y fusión

$$\mathbf{h}=(h_1 h_2 h_3 \dots h_9)^T ,$$

$$\text{Representa una homografía de la forma } \tilde{H} = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix}$$

**Formar la matriz A:** ensamblar las n (numero de correspondencias) matrices  $A_i$  , obtenidas de la etapa anterior y de dimensiones 2 x 9 cada una, en una única matriz  $A$  de dimensiones 2n x 9.

**Descomposición en valores singulares (SVD) de la matriz A:** SVD es un de los más potentes algoritmo de descomposición de matrices, particularmente para computaciones numéricas. Se suele utilizar para resolver sistemas de ecuaciones sobredeterminados (mas detalles sobre el metodo en [3] pag 585 ). El vector singular correspondiente con el valor singular más pequeño es la solución  $\mathbf{h}$ . Concretamente, si  $A=UDV^T$  con D diagonal con elementos positivos, ordenados de mayor a menor en la diagonal, luego  $\mathbf{h}$ , es la ultima columna de V.

**Determinar la matriz  $\tilde{H}$  a partir de  $\mathbf{h}$ :** una vez calculado el valor del vector  $\mathbf{h}$ , se obtiene la homografía que relaciona las correspondencias de puntos (normalizados) de la siguiente manera:

$$\mathbf{h}=(h_1 h_2 h_3 \dots h_9)^T \quad \tilde{H} = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix}$$

**Desnormalización:** En la etapa de normalización, los conjuntos de puntos fueron transformados utilizando dos matrices  $T$  ,  $T'$  . Para que la homografía calculada relacione las correspondencias según las coordenadas originales hace falta realizar renormalizarla. Esta desnormalización se realiza sobre la homografía calculada según la ecuación siguiente, donde  $T'$  y  $T$  fueron utilizadas en la etapa de normalización inicial.

$$H = T'^{-1} \tilde{H} T$$

#### 4.6.3. Implementación del algoritmo para conseguir el aumento de la resolución temporal

Como resultado del DLT, se obtiene una matriz H de transformación del espacio proyectivo  $\mathbf{P}^2$ . Es posible utilizar esta herramienta para unificar el plano proyectivo de varias imágenes, con algunas condiciones. Los planos (de la misma escena) capturados por cada una de las cámaras deben ser lo mas parecidos posible (referidos a zonas de la escena que aparecen en las imagen) y además ser estáticos, para evitar así que aparezcan defectos debido al cambio de perspectiva (La Ilustración 48 muestra un ejemplo de este hecho). Por este motivo las cámaras están situadas y acopladas a una estructura de

#### 4.Desarrollo de algoritmos de corrección de los errores asociados al proceso de captación y fusión

manera que queden lo mas juntas posible. Además la transformación únicamente es valida en el plano donde se encuentren la correspondencia de puntos. Si en la escena capturada hubiera objetos (estáticos o móviles) que se encuentran en distintos planos, habría que calcular la matriz de transformación para cada plano, algo que queda fuera del alcance de este proyecto.

En el caso de que la escena no se desarrolle en un único plano, hay que tener en cuenta las distancias entre los planos donde aparecen distintos objetos y la distancia entre la cámara y los mismos. Si la distancia entre los citados planos es pequeña con respecto a la distancia entre un plano (el más cercano a la cámara) y la cámara, el error cometido al considerar un único plano y realizar una única homografía puede ser inapreciable. Esta aproximación ha sido analizada en este trabajo para capturar escenas exteriores alejadas unos 30 metros del sistema de cámaras; la utilidad de esta herramienta en estos casos depende fuertemente de la composición de la escena capturada : objetos , fondo , planos ... .

Teniendo en cuenta estas condiciones, se ha desarrollado una implementación del algoritmo DLT para transformar los planos proyectivos de cada una de las cámaras a un plano proyectivo de referencia. Por conveniencia , este plano será considerado el de la primera imagen de la secuencia (o cámara) .

La organización lógica de la secuencia, explicada en el capítulo anterior, consiste en la intercalación de las imágenes capturadas por cada una de las cámaras que componen el sistema. Así, si escogemos las imágenes numero 2, 3 y 4 y sabemos que la secuencia fue capturada por dos cámaras, las imágenes 2 y 4 pertenecerán a la segunda cámara y la imagen 3 a la primera. Suponiendo que el plano proyectivo de las cámaras se mantiene fijo durante toda la secuencia, podemos unificar el plano proyectivo de las imágenes capturadas por el sistema de la siguiente manera:

- Detección de puntos correspondientes.
- Corrección de la distorsión radial (según se ha indicado en un apartado apartado 4.5).
- Determinación de la matriz H de transformación para cada perspectiva (apartado 4.6.2).
- Transformación de las imágenes.

##### *Detección de puntos correspondientes*

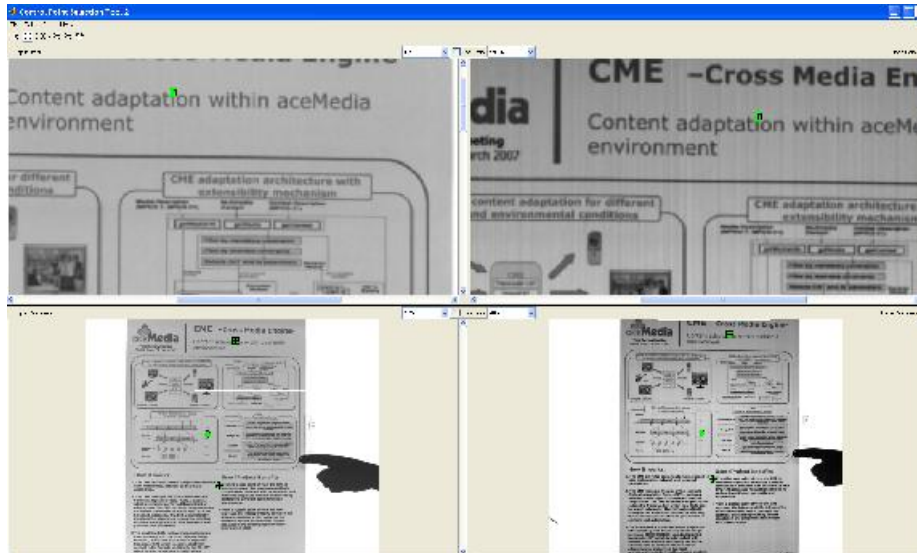
Como resultado de esta fase, esencial para el éxito del algoritmo implementado, se obtendrán correspondencias entre la imagen de referencia,  $x_i = (x, y)$ , y la imagen que deseamos transformar ,  $x'_i = (x', y')$ .

Dado que la selección precisa de puntos es crítica, la solución más razonable, ya que éste no es el objetivo de este proyecto, es la marcación manual de dichas correspondencias. En este proceso la precisión resulta determinante para la obtención de buenos resultados del método utilizado, de lo que el usuario debe ser consciente en todo momento. Además, tal y como fue expuesto anteriormente, el numero de

#### 4.Desarrollo de algoritmos de corrección de los errores asociados al proceso de captación y fusión

correspondencias debe ser mayor que cuatro, siendo recomendable unas seis o siete correspondencias distribuidas por toda lo imagen, si las hubiera.

MATLAB incluye, en el *Image Processing Toolbox*, una herramienta para la marcación manual de puntos correspondientes para dos imágenes. Esta herramienta se invoca con el comando *cpselect*, y resulta de gran utilidad para nuestro objetivo, Ilustración 47 .



**Ilustración 47 : Interfaz para la marcación manual de puntos y sus correspondientes**

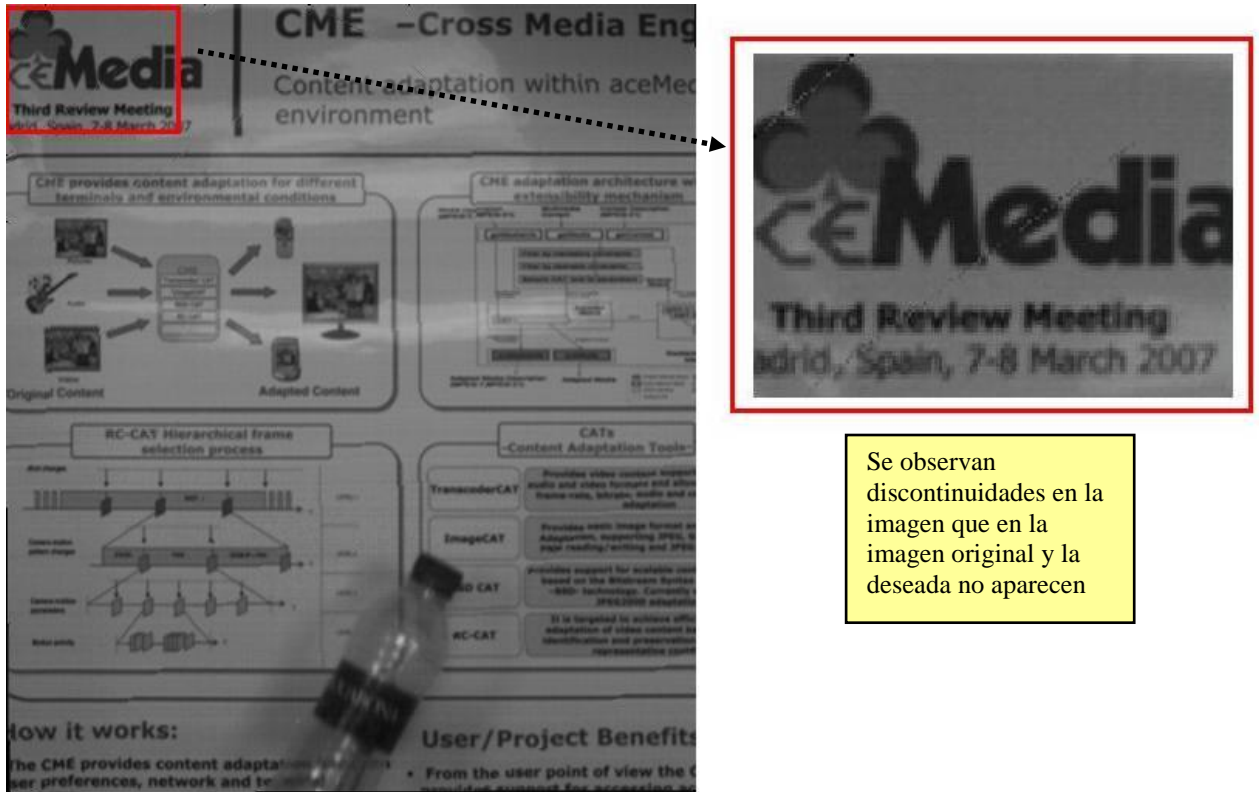
#### *Corrección de la distorsión radial*

La distorsión radial introduce no linealidades en el proceso de proyección, lo que provoca que la relación entre puntos homólogos deje de estar gobernada por una homografía. Las pruebas previas, que intentaban buscar la matriz H sin corregir la distorsión, producían resultados donde la imagen corregida aparecía dividida en zonas o regiones (tantas como puntos con correspondencias utilizados para el calculo de la homografía). Cada región, cuyo centro de masas era la coordenada original, era un recorte de la imagen original trasladada a ese punto. De esta manera la imagen modificada se asimilaba a un mosaico de recortes de la imagen original donde se cumplía la homografía calculada para los puntos utilizados en el cálculo de la misma. La Ilustración 48 , muestra un resultado de la transformación del plano proyectivo sin corregir previamente la distorsión radial.

Para corregir esta distorsión, se utiliza una matriz de correspondencia calculada previamente, según 4.5, a partir de los coeficientes que modelan la distorsión radial calculados para cada 'cámara'. La Ilustración 43 muestra las etapas resumidas del proceso que corrige la distorsión radial ejecutado para cámara. Como resultado de este algoritmo, se obtiene una matriz de correspondencia de puntos que indica las donde han sido desplazado cada punto de la imagen original (cada coordenada de la matriz corresponde a las coordenadas de imagen corregidas y el valor en la matriz indica las coordenadas de la imagen original de la que hay que obtener el valor de la intensidad). Utilizando esta matriz se realiza una interpolación utilizando las imágenes originales de la cámara.

#### 4.Desarrollo de algoritmos de corrección de los errores asociados al proceso de captación y fusión

Antes de pasar a la siguiente etapa, es necesario eliminar distorsión radial que presentan los puntos obtenidos en la etapa anterior. Utilizando los parámetros del modelo de distorsión (primer coeficiente y centro de la imagen) se aplica la formula para calcular las coordenadas corregidas.



**Ilustración 48: Resultado y detalle de la transformación del plano proyectivo de una imagen, utilizando homografías pero sin corregir la distorsión radial.**

#### *Determinación de la matriz H de transformación para cada perspectiva*

Utilizando la correspondencia entre puntos obtenidos en la segunda fase, se implementa el algoritmo DLT normalizado en transf\_perspectiva.m. Una vez obtenida la homografía **H** es necesario calcular la transformación que produce en todos los puntos de la imagen. Para ello se ejecuta una función que recorre todos los puntos uno a uno aplicando la ecuación:

$$x'_i = H x_i$$

donde  $x'_i$  representa las coordenadas homogéneas de la imagen corregida y  $x_i$  las coordenadas homogéneas de la imagen original.

El resultado de esta ecuación se va introduciendo en una matriz, denominada matriz de corrección, donde el valor de la coordenada (x,y) es el resultado de introducir en la ecuación los valores  $x_i = (x, y, 1)$ . Esta matriz será utilizada de la misma manera que la matriz de correspondencia calculada para corregir la distorsión radial.

#### *Transformación de las imágenes*

#### 4.Desarrollo de algoritmos de corrección de los errores asociados al proceso de captación y fusión

A través de la homografía  $H$ , y aplicando directamente la ecuación  $x'_i = Hx_i$  para cada punto en la imagen, se obtiene una matriz de correspondencia, de la forma

Matriz de correspondencia  $(x, y) = (x' y' w')$  donde  $w' = 1$

Queda de la forma

Matriz de correspondencia  $(x, y) = (x' y')$

Esta matriz representa de dónde proviene el valor de intensidad de los pixeles de la imagen corregida. Por ejemplo, si el valor en la coordenada  $(4, 5)$  de esta matriz es  $(4.6, 7.5)$ , esto indica que el valor de intensidad en el punto  $(4,5)$  en la imagen corregida es el valor en el punto  $(4.6,7.5)$  de la imagen original.

Haciendo una interpolación de segundo orden, de la misma manera que en la corrección radial (ver Ilustración 42), se obtiene la imagen con el plano proyectivo corregido. Una vez ha sido obtenida la matriz de correspondencia para cada una de las cámaras, es posible corregir la perspectiva de todas las imágenes de la secuencia sin necesidad de volver a calcular los valores de la matriz de transformación  $H$ .

#### 4.6.4. Implementación del algoritmo para conseguir el aumento de la resolución espacial

Como secuencia de partida se tienen  $n$  imágenes ( $n$  es el número de cámaras) para cada instante de tiempo. Cada una de estas imágenes ha sido capturada por una cámara distinta que están separada de las demás una distancia dependiente de algunas características de la escena: distancia al objetivo y óptica utilizada.

Estas imágenes necesitan tener una región común a la o las imágenes vecinas (la/s de la cámara más próxima) para poder encontrar correspondencias, fusionar las imágenes y componer la imagen panorámica en cada instante. Tras varias pruebas preliminares, esta región común se ha fijado al 10% de la imagen y se obtiene de uno de los extremos laterales de cada imagen.

Las imágenes de los extremos del sistema, tan solo tendrán en común con otras imagen un lateral. Sin embargo, las imágenes 'centrales' compartirán las regiones de ambos laterales con otras imágenes. Si la resolución horizontal de la imagen de cada cámara es  $H$  y se utilizan  $n$  cámaras del sistema, la imagen panorámica resultante tendrá una resolución de

$$\text{Resolución de la imagen panorámica} = H*n - (0.1*H*(n-1))$$

Estos valores asumen que el alineamiento de las cámaras es óptimo. Esta resolución puede variar dependiendo del alineamiento de las cámaras.

Inicialmente, la composición de la panorámica se hizo de manera manual. Aún así los resultados no eran del todo malos si el usuario tenía la suficiente paciencia para ubicar las cámaras correctamente, ajustar el enfoque y apertura del diafragma. Para



#### 4.Desarrollo de algoritmos de corrección de los errores asociados al proceso de captación y fusión

evitar esta dependencia del usuario para la ubicación y colocación de las cámaras, se pueden utilizar transformaciones del plano proyectivo para fusionar imágenes y componer la imagen panorámica de una manera más precisa y automatizada. Con esta herramienta aumentamos la autonomía del algoritmo que compone la imagen panorámica. Las etapas realizadas para lograr el objetivo son:

- Detección de puntos correspondientes
- Corrección de la distorsión radial (según se ha indicado en un apartado apartado 4.5).
- Determinación de la matriz  $H$  de transformación para cada perspectiva ( apartado 4.6.2)
- Transformación de las imágenes

##### *Detección de puntos correspondientes*

Esta etapa resulta determinante para la obtención de resultados. Al igual que en el algoritmo de transformación del plano proyectivo utilizado para aumentar la resolución temporal, se utilizará un asistente para la marcación de puntos homólogos en dos imágenes. La Ilustración 47 representa una captura de la interfaz utilizada.

Para cada pareja de imágenes (de cámaras vecinas), hay que marcar un mínimo de 6 correspondencias en la zona común para poder calcular la homografía que las define. Estas correspondencias deben estar lo mas distribuidas posible por toda la región común y no deben estar alineadas.

Por este motivo, se utiliza un 10% de la dimensión de la imagen (124 píxeles para los modelos de cámaras Y-3) para representar la región común. A su vez, al tratarse de zonas bastante pequeñas es necesario que haya en las mismas las suficientes referencias para encontrar las correspondencias con las características citadas, lo que establece ciertas restricciones sobre el fondo de la escena.

##### *Corrección de la distorsión radial*

En este modo resulta determinante la distorsión radial. La distorsión radial es mas acentuada cuanto más se aleja del centro de la imagen. Puesto que las correspondencias se obtienen en los márgenes de las imágenes, antes de proceder a calcular las homografías que definen las correspondencias de puntos (resultado de la etapa anterior) es necesario corregir la posible distorsión radial.

Además de utilizar la matriz de correspondencia calculada en el proceso de calibración de cada una de las cámaras, hay que corregir las coordenadas de los puntos obtenidos en la etapa anterior. Esta corrección resulta necesaria para calcular la homografía que relaciona ambas imágenes a partir de puntos sin distorsión radial.

##### *Determinación de la matriz $H$ de transformación para cada perspectiva*

Utilizando una implementación del algoritmo DLT se obtiene la homografía  $H$ , que define la relación entre las coordenadas de los puntos homólogos (sin distorsión radial).

#### 4.Desarrollo de algoritmos de corrección de los errores asociados al proceso de captación y fusión

$$x'_i = H x_i$$

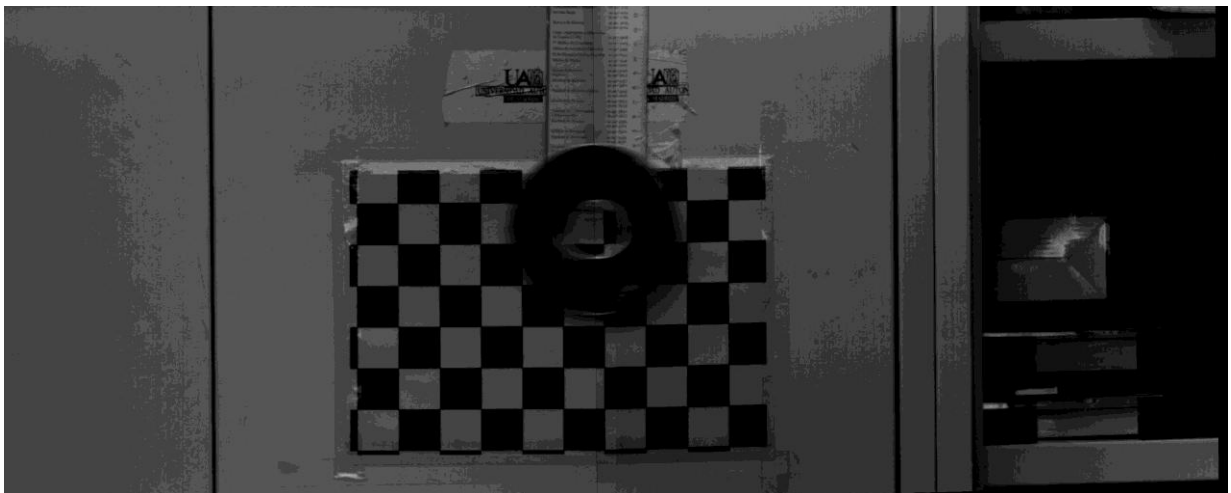
Utilizando esta fórmula, se calcula la matriz de correspondencia que indica el desplazamiento sufrido por cada punto de la imagen original. Las dimensiones de esta matriz serán las máximas posibles para la imagen panorámica calculada (en un sistema con dos cámaras de 1024x 1240 de resolución, la matriz tendrá una dimensiones de 1024 x 2480 píxeles).

##### *Transformación de las imágenes*

Para la composición de la imagen panorámica, se ha de tomar una de las cámaras como referencia. Para este caso la imagen de la cámara 1 (la ubicada mas a la izquierda en el sistema) será tomada como referencia. Esta primera imagen no modifica su plano proyectivo y se utiliza como referencia en el proceso de ajuste de contraste. La única transformación que sufrirá esta imagen es la corrección de la distorsión radial.

Al realizar la interpolación utilizando la matriz de correspondencia (que corrige la perspectiva) y una imagen de la cámara siguiente, hay que tener en cuenta que esta imagen será desplazada hacia el lateral derecho de la panorámica y la imagen de referencia estará ubicada en la zona izquierda de la panorámica.

La Ilustración 49 muestra el resultado de éste algoritmo utilizando dos imágenes. En esta imagen ha sido ajustado el contraste global, tal y como se expone en el apartado 4.7.3 , pero no así la posible distorsión radial de las imágenes individuales de las que se partió. Han sido utilizadas 12 correspondencias para calcular la homografía.



**Ilustración 49 :** Imagen panorámica compuesta utilizando 2 imágenes y una única homografía. Se ha ajustado el contraste .

#### **4.6.5. Conclusiones y trabajo futuro**

La geometría proyectiva es un conjunto de técnicas de carácter geométrico que permite representar el espacio tridimensional sobre una superficie bidimensional. En nuestro proceso de captura cada cámara hace una representación única de la escena que depende de su ubicación relativa.

#### 4.Desarrollo de algoritmos de corrección de los errores asociados al proceso de captación y fusión

En este apartado se expone un método y una implementación para unificar la perspectiva de las ‘cámaras’ que componen el sistema. Bajo ciertas condiciones (que la escena se desarrolle en un mismo plano), estas imágenes están relacionadas mediante una transformación 2D-2D, denominada homografía.

En el modo *AumentoCampo*, se utilizan homografía para componer imágenes panorámicas y no tener que confiar en la habilidad del usuario para colocar las cámaras y así conseguir la panorámica sin ningún tratamiento post-captura.

Existen varios métodos para el cálculo de homografías entre parejas de imágenes a partir de correspondencia entre puntos y líneas. Para este PFC hemos elegido el algoritmo DLT normalizado que produce muy buenos resultados si se parte de una buena cantidad de correspondencias (unas 6-8 bien distribuidas por toda la imagen).

Para verificarlo, se ha implantado el algoritmo a nuestro objetivo: aumento de la resolución temporal y espacial. Para ello se han ideado distintas etapas: marcación de puntos característicos, corrección de la distorsión radial, cálculo de  $n-1$  ( $n$  numero de planos proyectivos, puntos de visión, cámaras...) matrices de transformación e interpolación de segundo orden.

En ambos objetivos las etapas de marcación de puntos característicos, corrección de la distorsión radial y cálculo de la homografía utilizando DLT son comunes y no existen diferencias. Ambos algoritmos difieren en la ultima etapa, que en el caso del aumento de la resolución temporal se trata de una transformación del plano (de cada cámara menos la de referencia) para conseguir la misma perspectiva en todas las cámaras. Y en el segundo, se transforma el plano proyectivo (de cada cámara menos la de referencia) para conseguir una imagen panorámica.

#### *Trabajo Futuro*

En cuanto al trabajo futuro, es posible mejorar la implementación y el método en varios campos. El más prioritario es la detección de correspondencia entre puntos característicos de manera automática; sin embargo se trata de técnicas costosas y con una fiabilidad baja en las condiciones tan poco restrictivas (exteriores con fondos no controlados) que considera este proyecto. Suponiendo que es posible detectar tantas correspondencias como deseamos, una segunda mejora seria aplicar estas transformaciones a imágenes donde hubiera más de un plano dentro de la escena capturada. Esta mejora daría la posibilidad de aumentar la resolución espacial de escenas tridimensionales, en vez de obligar, como ahora ocurre, a que la escena grabada se desarrolle dentro de un mismo plano.

### **4.7. Ajuste de contraste**

En este apartado, se analizan las distintas técnicas para ajustar el contraste, nivel de luminancia, de las imágenes capturadas en los dos casos estudiados en este proyecto: aumento de resolución temporal y espacial. Se hace un rápido recorrido por distintas técnicas implementadas y probadas, y se propone una solución robusta para ambos casos sin tener en cuenta tiempos de ejecución.

### **4.7.1. Técnicas de ajuste de contraste**

Existen varias técnicas, transformaciones, para el ajuste de la intensidad de los píxeles en imágenes. Es posible separar dichas técnicas en dos grandes grupos: aquellas donde abordan en tema desde el dominio espacial y aquellas que lo abordan desde el frecuencial.

La primera se basa en la modificación directa del valor de los píxeles en la imagen, principalmente para modificar la intensidad de los píxeles, realizar filtrado espacial, umbralizar la imagen, ajuste e igualación de histogramas...

La segunda utiliza la representación de la imagen en un histograma para ajustar el mismo a una forma deseada.

A partir de aquí hay que separar de nuevo el objetivo de nuestro proyecto y analizar la situación en cada caso.

### **4.7.2. Ajuste de contraste para el aumento de la resolución temporal**

En el objetivo del aumento de la resolución temporal. Se tienen hasta 4 fuentes de imágenes capturando la misma escena, cada una con un contraste y enfoque distintos; además la iluminación en los objetos sólidos es desigual debido al cambio de perspectiva de las cámaras.

Una transformación global, por ejemplo un filtrado (en cualquiera de los dos dominios), modifica la imagen de manera global. En este caso, es necesario que los píxeles sean modificados directamente por su intensidad, ya que la diferencia de iluminación entre las imágenes varía de manera local.

Se supone que todas las imágenes han unificado su perspectiva (transformaciones del plano proyectivo) y además ha sido corregida la posible distorsión radial introducida por la lente utilizada en cada cámara. Bajo estas condiciones es posible suponer que cada píxel una imagen corresponde con el mismo en el resto de las imágenes, siempre y cuando en este píxel se represente un objeto estático en la escena denominado fondo. Es decir, los píxeles que representan el fondo de la escena deberán tener todos, en todas las imágenes, valores similares. Una pequeña diferencia (mayor que cuatro, por ejemplo, en una escala de 255 niveles) del valor de estos píxeles entre imágenes produce un parpadeo del fondo de la imagen al visualizar la secuencia.

Por este motivo se tiene que diferenciar entre objetos móviles y fondo, o *background*, en cada una de las imágenes que componen la secuencia y considerar la composición de las imágenes vecinas. Esta separación resulta necesaria ya que su contraste debe ser ajustado de diferente manera. Además cada imagen ajustada debe ser lo mas homogénea posible con respecto a las imágenes 'vecinas' sin deteriorar la calidad, en especial la de los objetos móviles.

Para igualar las intensidades de los píxeles que pertenecen al fondo, se hace una media de los valores de intensidad de esos píxeles en las imágenes que componen un

#### 4.Desarrollo de algoritmos de corrección de los errores asociados al proceso de captación y fusión

ciclo de grabación. Esta operación puede difuminar la imagen debido a las imperfecciones de las etapas de corrección de distorsión radial y unificación de la perspectiva pero consigue ajustar con eficacia el contraste de esta zona de manera que se asimile a las imágenes vecinas (que han sido capturadas por cámaras distintas). Las regiones de movimiento deben ser tratadas de manera que no se pueda perder resolución en la operación de ajuste de contraste, ya que se desea realizar medidas sobre los objetos móviles de la escena.

Para distinguir el fondo del objeto, se restan las imágenes anterior y posterior de la secuencia que capturó la cámara a la que corresponde la imagen que se quiere ajustar. En esta diferencia los máximos corresponden con regiones que han variado su intensidad considerablemente: zonas con movimiento o/y ruido. Se renormaliza la imagen (para que los valores de la imagen ocupen todo el rango de valores posible, de 0 a 255), se ejecuta un operador morfológico de cierre (para eliminar el ruido) y se umbraliza para separar la parte móvil del fondo. Con estas operaciones se obtiene una máscara que aplicada sobre la imagen distingue fondo y *objeto* (o al menos las fronteras de los objetos móviles)

El ajuste de contraste del fondo de la imagen se obtiene calculando la media de los fondos de las  $n$  (número de cámaras) imágenes que componen el ciclo de la secuencia; éste ciclo es considerado desde que tratamos una imagen capturada por una cámara hasta que vuelve a tratarse la siguiente imagen capturada por la misma .

Para igualar el contraste de la parte móvil, no es posible calcular una simple media ya que aparecería una estela, sombra, que difuminaría la trayectoria del objeto móvil y reduce la precisión en su localización. Por este motivo el objeto móvil de la imagen debe ser tratado de manera independiente. Es necesario conocer cuánto varía la intensidad (diferencia entre el fondo de la imagen original con respecto al fondo ajustado) en las vecindades de la región móvil. Para conocer estos valores, se recorren píxel a píxel la región móvil. Para cada píxel se calcula una media de la variación de los vecinos que pertenecen al fondo, unos 20 píxeles. Luego este valor se almacena y se pasa al siguiente píxel. Se tiene una matriz con los valores de la diferencia de intensidad entre la imagen original y la imagen contrastada con respecto a sus vecinas. Una vez terminada esta etapa, se suma esta matriz (cada valor es calculado según la categoría del píxel, fondo u objeto) con la imagen a tratar. La Ilustración 50 muestra un ejemplo de este proceso para una imagen de una secuencia compuesta a partir de 2 cámaras. En este proceso de ajuste, debido a la necesidad de utilizar información de las imágenes vecinas, no son tratadas las primeras y últimas imágenes capturadas por cada cámara, por lo que no se ajustan la intensidad de 4 imágenes, en un caso más general  $2*n$  (donde  $n$  es el número de cámaras).

#### 4.Desarrollo de algoritmos de corrección de los errores asociados al proceso de captación y fusión

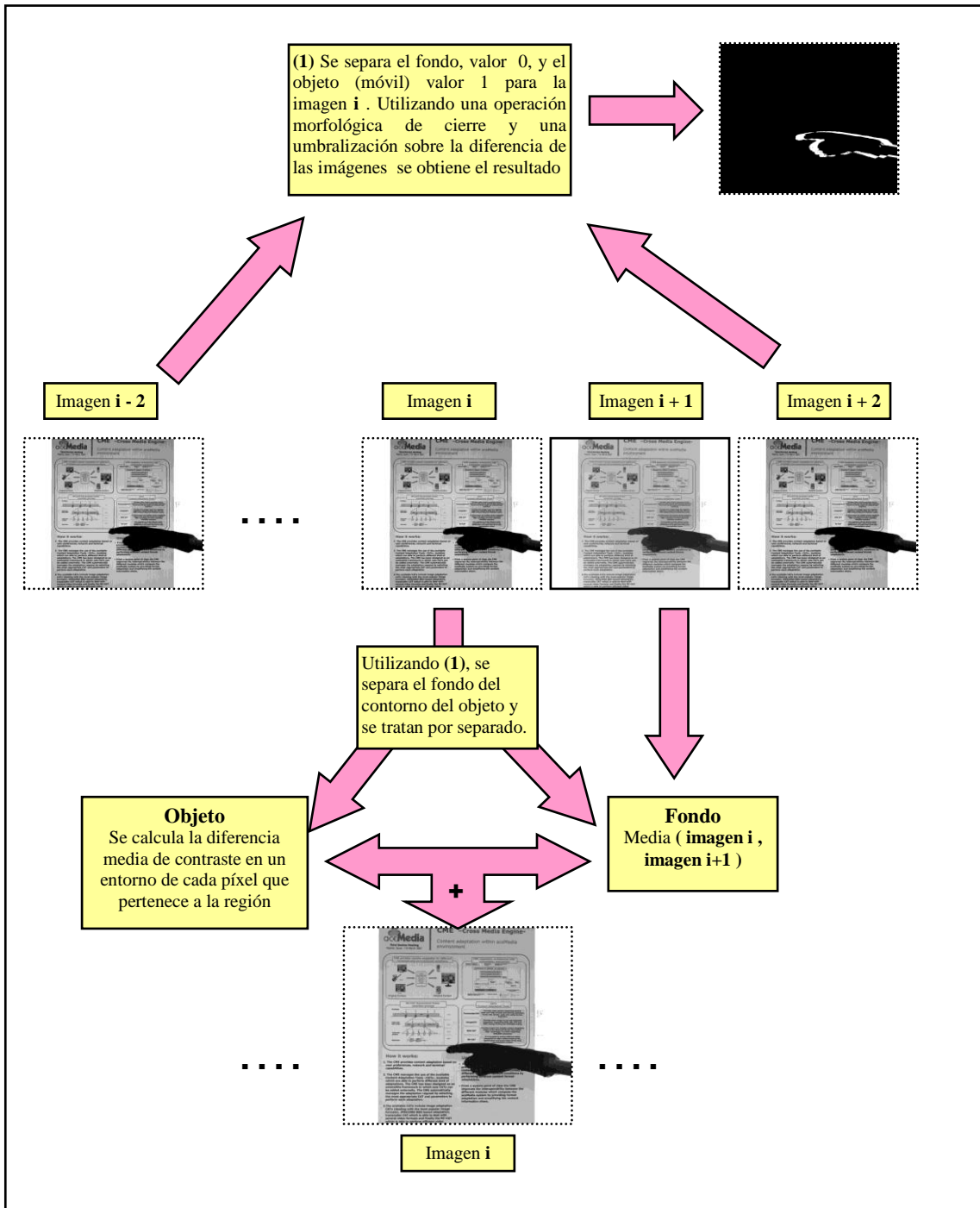


Ilustración 50 : Esquema de ajuste de contraste para el modo *UltraSpeed*

#### 4.7.3. Ajuste de contraste para el aumento de la resolución espacial

Al igual que en el modo anterior, se tienen hasta cuatro fuentes de imágenes con distintas ópticas (enfoque y apertura de obturador). En este modo, el objetivo es construir una imagen panorámica a partir de las imágenes individuales capturadas por cada cámara. La imagen panorámica que resulta en un instante de tiempo es una imagen

#### 4.Desarrollo de algoritmos de corrección de los errores asociados al proceso de captación y fusión

compuesta a partir de la fusión de las imágenes originales capturadas en ese instante por cada cámara. Esta fusión se consigue utilizando homografías (calculada a partir de la región solapada de las imágenes) y corrigiendo la distorsión radial de cada una de las cámaras (que depende de la óptica utilizada por cada cámara).

Una vez obtenida la imagen panorámica resultado del proceso de fusión, se observa en las zonas comunes una sobre-iluminación (resultado de la suma de las dos imágenes) además de una diferencia de contraste entre las zonas no comunes de las imágenes. Ésta diferencia entre dos imágenes contiguas depende de la iluminación de la parte de la escena capturada y de la configuración de la lente de cada cámara (enfoque y apertura del obturador). El ajuste realizado, se centra en compensar la diferencia de contraste entre imágenes contiguas producidas por la diferencia de configuración de los sistemas ópticos utilizados por cada cámara .

La idea de ajuste de contraste en este modo consiste en que no aparezcan discontinuidades del valor de los píxeles muy acentuadas en la imagen final debido al proceso de fusión. Es en la región común, utilizada para fusionar las imágenes individuales, donde más se aprecia la diferencia de contraste entre las imágenes.

Para conseguir una distribución homogénea del contraste de la imagen panorámica se toma una imagen individual (cámara) como referencia. El resto de las imágenes (cámaras) modificaran su valor medio de manera que quede ajustado a la imagen de referencia. El valor calculado se aplica, sumándolo a las intensidades de los píxeles de toda la imagen, de manera global para no perder resolución.

En la región común cada píxel tendrá una intensidad resultado del cálculo de la media de las intensidades de las dos imágenes. Esta región la determina la transformación del plano proyectivo realizado sobre la ‘segunda’ imagen y representa la región de solape entre ambas imágenes. Al realizar esta media, los valores de los píxeles en la región no deben variar mucho con respecto a la imagen de referencia. Para ello los valores de intensidades en esta región de la segunda imagen deben ser lo mas próximos posible a los de la imagen de referencia sin modificar la resolución. Se calcula la diferencia media de los píxeles entre la imagen de referencia y la imagen contigua en la zona común. Y se aplica (se suma) este valor sobre las intensidades de la imagen original contigua. De esta manera la diferencia de contraste es ajustada antes de realizar cualquier otra operación, de modo que en las regiones comunes la intensidad media de los píxeles de cada imagen es idéntica.

A continuación, se realizan las interpolaciones sobre la imagen de la segunda cámara que corrigen la distorsión y la perspectiva y se compone la secuencia panorámica, tal y como se describe en el apartado 4.6.4

#### **4.7.4. Conclusiones**

Al visualizar una secuencia capturada por un sistema (cualquiera de los dos modos) donde no hay una configuración similar de las ópticas utilizadas en cada cámara, se tiene la percepción de que la secuencia tiene una calidad baja. Con las operaciones descritas se consigue ajustar el contraste de cada imagen con respecto a sus

#### 4.Desarrollo de algoritmos de corrección de los errores asociados al proceso de captación y fusión

vecinas (las imágenes capturadas por las otras cámaras) y aumentar la percepción de calidad de la secuencia final. Aunque en la realidad, con estas operaciones se pierde definición en ciertas regiones de las imágenes capturadas originales (ya lejos quedan de serlo puesto que previamente se ha realizado sobre ellas una interpolación para modificar su perspectiva y la distorsión radial).

La etapa de ajuste de contraste se ejecuta en distinto orden dependiendo del modo. Para el modo *AumentoTasa* , este ajuste se realiza una vez finalizado el proceso de corrección de la distorsión radial y la transformación del plano proyectivo. Para el modo *AumentoCampo* , en cambio , este proceso se ejecuta antes de realizar cualquier transformación.

En el caso del aumento de la resolución temporal, las imágenes contrastadas deben tener una intensidad muy aproximada en las zonas donde no se produzca movimiento. Las zonas móviles deben ser ajustadas de manera que no se pierda la nitidez de la imagen original. Por este motivo, en cada imagen es necesario separar fondo y objeto (parte móvil). Esta separación resulta fundamental para poder obtener resultados que mejoren la calidad, subjetiva sin deteriorar la objetiva, de la secuencia original.

Este ajuste en este modo es sólo aplicable si las imágenes de la secuencia original tienen una única perspectiva; en este documento se refiere a que previamente han sido corregidas la perspectiva y la distorsión radial.

Para ajustar el contraste en el segundo objetivo, se toma una cámara como referencia y se ajusta de manera global el resto para que la distribución del contraste en la imagen panorámica sea lo más homogénea posible. Para ajustarlo se utiliza información de las zonas comunes. En esta zona se calcula la diferencia media entre las intensidades de los píxeles de cada imagen y se aplica este valor sobre toda la imagen original (la de la segunda cámara o demás). A continuación se corrige la distorsión radial, perspectiva y se compone la imagen panorámica. En este modo el proceso de ajuste de contraste se realiza antes de componer la imagen panorámica, sobre las imágenes originales (todas menos la referencia). De este modo se consigue que la iluminación sea homogénea en las regiones solapadas sin interferir en la resolución del resto de la imagen.

### **4.8. Resultados y conclusiones**

En este capítulo se ha presentado la problemática asociada a la obtención de la secuencia final en cada modo, fusión (*AumentoCampo*) e intercalación de imágenes (*AumentoTasa*). Muchos son los factores que producen que la secuencia resultante no sea óptima, pero algunos resultan más determinantes que otros.

Para el aumento de la resolución temporal, el principal problema es el cambio de perspectiva debido a la ubicación de las cámaras. Hay que unificar la perspectiva y para ello se utilizan técnicas de transformación del plano proyectivo, homografías. Un ajuste de contraste aumenta notablemente la calidad subjetiva de la secuencia. A pesar de perder resolución, este proceso , resulta necesario.



#### 4.Desarrollo de algoritmos de corrección de los errores asociados al proceso de captación y fusión

Para conseguir una imagen panorámica, a partir de las imágenes capturadas por el sistema en modo *AumentoCampo*, es necesario registrar las imágenes. Es decir, encontrar correspondencias en las regiones comunes para poder fusionarlas. Además de corregir la distorsión radial y ajustar el contraste global de la imagen panorámica.

Para intentar mejorar la calidad de la secuencia resultante en el modo *AumentoTasa*, se han implementado una serie de herramientas que subsanan los errores introducidos por las limitaciones del sistema utilizado. Las etapas propuestas en la Ilustración 27, han sido desarrolladas en funciones de Matlab que se encuentran ubicadas en la carpeta \Correcciones. En esta carpeta el *scrip* test.m ejecuta cada una de las etapas: corrección de la distorsión radial, homografías y ajuste de contraste.

Para observar alguna aplicación práctica, se puede visualizar la evolución de la secuencia resultante en la URL [http://www-gti.ii.uam.es/publications/PFC EJS](http://www-gti.ii.uam.es/publications/PFC_EJS)

#### *Consideraciones*

Hay que tener en cuenta que el objetivo de este proyecto es la precisión de la secuencia resultante. Es por ese motivo por lo que no se ha tenido mucho en cuenta el tiempo de ejecución de los algoritmos implementados; de todas formas, la carga computacional que conllevan las herramientas desarrolladas es alta debido a la fundamentación teórica que las respalda. De vista a reducir este tiempo de ejecución, existe una operación común en dos de las herramientas desarrolladas: calculo de la matriz de correspondencia e interpolación. La ‘doble’ interpolación introduce un error, que es posible reducir si se integran ambas operaciones en una sola. En este aspecto, ha sido desarrollada una función (*transf\_perspectiva\_y\_radial\_distortion.m*) que integran ambos algoritmos y cuyo resultado es una única matriz de correspondencia, y por consiguiente una única interpolación.

Estos algoritmos están desarrollados para utilizarse con imágenes en escala de grises. Para la aplicación en secuencias a color, estos procesos se hacen en cada una de las bandas que componen una imagen a color y por consiguiente el tiempo de ejecución se multiplica por tres.

#### *Trabajo futuro*

Para poder realizar una valoración real de la utilidad de los algoritmos implementados, sería necesario añadir a la aplicación desarrollada (manejo y control de las cámaras), una interfaz que diera la posibilidad al usuario de ejecutar uno o varias de estas herramientas sobre la secuencia obtenida. Además, de generar una secuencia resultado de estas transformaciones. También, es posible mejorar el resultado en el modo panorámico utilizando las herramientas desarrolladas en este capítulo pero éste ya sería trabajo para otro proyecto.

## 5.Conclusiones

---

En este proyecto llevado a cabo, se pone en práctica la utilidad de los sistemas múlticámara. La idea general desarrollada es que utilizando varias cámaras es posible mejorar la tasa de cuadros y la resolución espacial de las secuencias capturadas individualmente por cada una de las cámaras. La calidad de la secuencia final está muy determinada por la ubicación de las cámaras, sistemas ópticos utilizados, configuración de los parámetros de cada una de las cámaras, sincronismo ...

La secuencia final debe asimilar que ha sido capturada por una única cámara además debe tener la suficiente calidad para que puedan ser tomadas medidas sobre los objetos que aparecen en ella. Por este motivo es necesario aplicar sobre la secuencia final una serie de algoritmos para subsanar los defectos naturales del sistema.

En la primera fase del proyecto, se diseñó un sistema compuesto de dos a cuatro cámaras de óptica fija. Además según el objetivo deseado, aumento de la resolución espacial o temporal, la configuración de los parámetros (tiempo de exposición y periodo de captura) y el sincronismo debía ser determinado de manera diferente.

En una segunda fase, se desarrollaron una aplicaciones propias para la configuración, alineamiento de las camaras, captura y montaje de la secuencia final. Como resultado de esta etapa, se obtuvieron para cada modo dos aplicaciones relacionadas:

- La primera permite al usuario configurar algunos parámetros de las cámaras (*binning* , *ganancia analógica* y *digital*) además de visualizar cada cámara por separado , habilitar/deshabilitar las cámaras e implementar un algoritmo de alineamiento .
- La segunda aplicación, permite configurar parámetros relativos a la escena a capturar (tiempo de exposición, periodo de captura , numero de imágenes por segundo ... ) . Las cámaras tienen una memoria interna donde almacenan la secuencia capturada, el usuario puede seleccionar qué parte de la secuencia quiere descargar al ordenador. Utilizando un reproductor donde se visualiza la secuencia final montada pero sin tratamiento alguno , el usuario selecciona la imagen inicial y final que descargará a disco.

Además de estas aplicaciones, se desarrolló una aplicación para realizar análisis cinemático de secuencias de imágenes. En esta aplicación, a partir de puntos marcados por un usuario, se calculan medidas físicas (velocidad, aceleración lineal y angular ...) y se presentan en un archivo EXCELL.

Por ultimo, una vez obtenidas secuencia capturadas con el sistema de cámaras de cada uno de los modos. Se realizaron algunos algoritmos para la mejora de la secuencia final. A grandes rasgos, los principales problemas de la secuencia montada eran el cambio de perspectiva y diferencia de contraste entre las imágenes de cada una de las cámaras. Para corregir el cambio de perspectiva se utilizaron homografías que resultaron ser incompletas para generar buenos resultados. Las imágenes generadas tenían una distorsión radial (producida por la lente utilizada en cada caso) que había que corregir antes de modificar la perspectiva. Para el ajuste de contraste, se realizaron

operaciones básicas entre imágenes en un orden lógico para que fueran lo más genérico posible.

### ***Trabajo futuro***

En este proyecto se inician varios temas donde es posible mejorar y mucho los resultados obtenidos en este proyecto. A continuación se enumeran cada una de las etapas discurridas y se exponen las posibles mejoras que se pueden implementar . Aunque para entender las lineas con ams detalles consultar el apartado de conclusiones de cada capitulo en cuestión.

- Etapa de desarrollo de aplicaciones para el manejo de un sistema de cámaras de alta velocidad: esta aplicación únicamente funciona con algunos modelos de cámaras de la empresa IDT. Futuros trabajos podrían añadir la funcionalidad para otros modelos de cámaras y otros fabricantes. Intentar mejorar la velocidad de ejecución de la aplicación desarrollando esta aplicación en otro lenguaje más competitivo en este aspecto.
- Etapa de desarrollo de herramientas para mejora de la secuencia final: En esta etapa existen varios temas iniciados donde se podría mejorar. El primero, mas estudio sobre la distorsión radial y la extracción de los parámetros que la definen a partir de plantillas de imágenes. La segunda, utilizar homografías para corregir la perspectiva de escenas que se desarrollen en más de un plano.

## **Referencias**

---

- [1] "MotionPro X Camera SDK Reference Manual Software Development Kit"
- [2] "MotionPro X Timing Hub SDK Reference Manual Software Development Kit"
- [3] "Multiple View Geometry in computer vision", Richard Hartley and Andrew Zisserman, Cambridge University Press,
- [4] "The Image processing Handbook". John C.Russ . CRC PRESS
- [5] "Digital Image Processing" . Rafael C. Gonzalez and Richard E. Woods. Pearson
- [6] "Visión por computador Imágenes digitales y aplicaciones " Jesús M. de la Cruz Gonzalo Pajares Editorial Ra-Ma
- [7] "A flexible new technique for camera calibration. *IEEE Translations on Pattern Analysis and Machine Intelligence*". Zhang Z.(2000)
- [8] "The geometry of multiple images " OLIVIER FAUGERAS NAD QUANG-TUAN LUONG MIT Press

## A. Manual de Usuario

---

### A.1. *Introducción*

En este apartado se describe la funcionalidad de la aplicación desarrollada, teniendo en cuenta las especificaciones iniciales además de las limitaciones del sistema.

Parece razonable idear una única solución para resolver ambos problemas ya que los dos comparten muchísimas similitudes. Pero para una mayor organización y simplicidad, se prefirió solucionar de manera individual los dos problemas planteados de modo que el resultado son dos sub-aplicaciones separadas, con bastantes partes comunes pero con pequeñas diferencias: disposición de los objetos en el marco, configuración del modo de grabación, algoritmo de alineamiento...

A su vez en cada una de estas sub-aplicaciones o *modos* de captura se definen distintas etapas para la obtención de la secuencia deseada

- Configuraron y alineamiento de las cámaras
- Captura y descarga de la secuencia a disco.
  
- Además se añade una aplicación para el análisis cinemático de la secuencia resultante, que a diferencia de los dos puntos previos, sería común a ambos modos ya que se puede utilizar con cualquier secuencia de imágenes,

Al tratarse de una aplicación con fines comerciales y desconocer el tipo de usuario que lo va a utilizar, se pone especial énfasis en desarrollar una aplicación sencilla de manejar, con un entorno *amigable* y la completa funcionalidad que permite el SDK de la cámaras utilizadas.

Debido a la similitud entre ambas aplicaciones, a continuación se analiza la aplicación implementada para el modo *AumentoTasa* y posteriormente se exponen los cambios introducidos para desarrollar el modo *Panorámic*. Por ultimo se exponen los principios de la aplicación destinada al estudio físico de secuencia, denominado *Cinematic Analysis*

### A.2. *Ventana de inicio*

Para ejecutar ambos modo se opto por implementar una ventana de inicio , Ilustración 18 , donde el usuario elige el modo de ejecución, antes de iniciar el modo correspondiente debe estar montado el sistema de acuerdo al mismo. Esta aplicación, programada en *Inicio.m* y *Inicio.fig* , muestra la siguiente interfaz : Ilustración 51

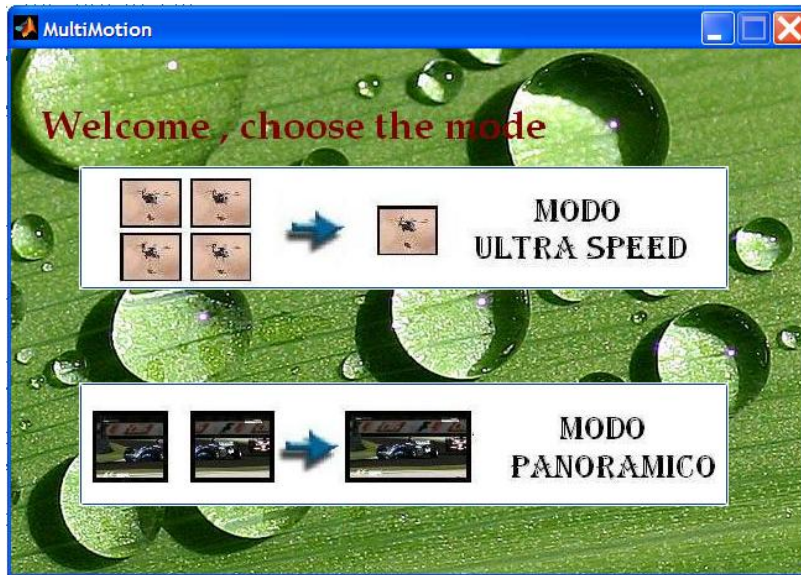


Ilustración 51 : Ventana de Inicio

### A.3. Modo AumentoTasa: Asistente de alineamiento y ajuste

Esta aplicación ha sido desarrollada para configurar los parámetros del sistema de cámaras: tiempo de exposición, *binning*, ganancia del sensor... Además se utiliza para indicar el orden que seguirá la secuencia al componerla y realizar una alineación de las cámaras. Esta alineación consiste en el calculo de la región de interés común a todas las cámaras implicadas en el proceso de captura, consultar capítulo 4.4.2.

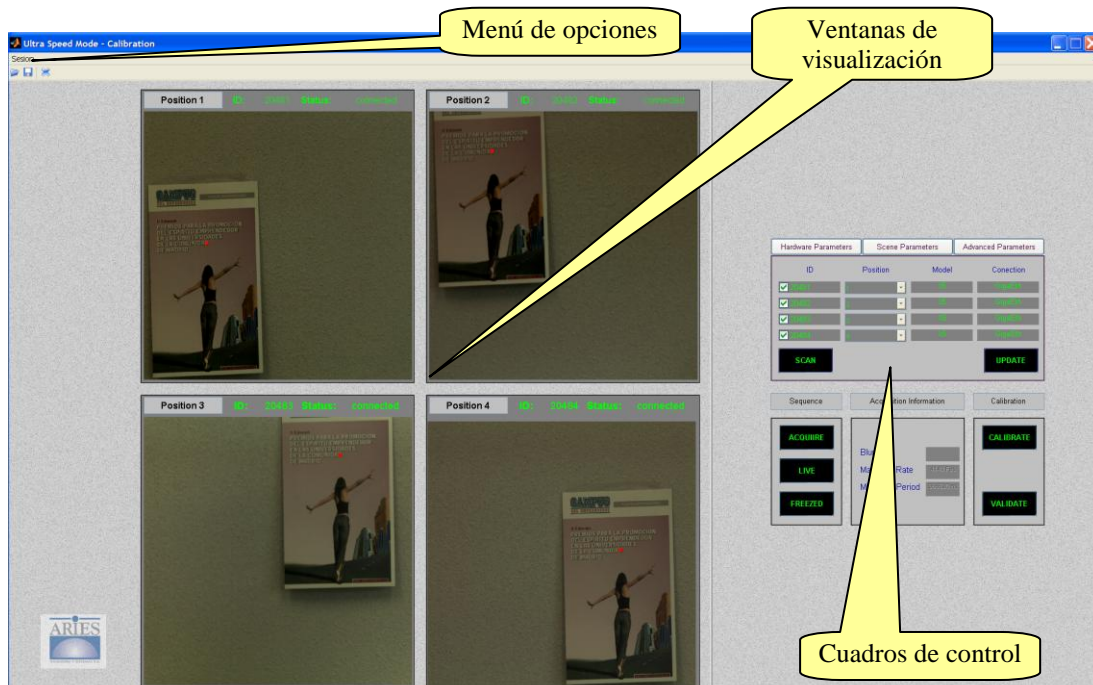


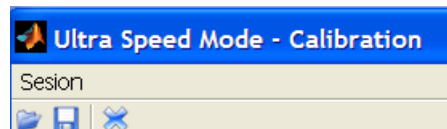
Ilustración 52 : Aspecto general del interfaz de ajuste y alineamiento para el modo *AumentoTasa*

Para el desarrollo de este fase se ha separado en tres partes la interfaz,

- Menú de Opciones
- Ventanas de visualización
- Cuadro de controles

A continuación serán descritas la funcionalidad y características de cada uno de estas partes.

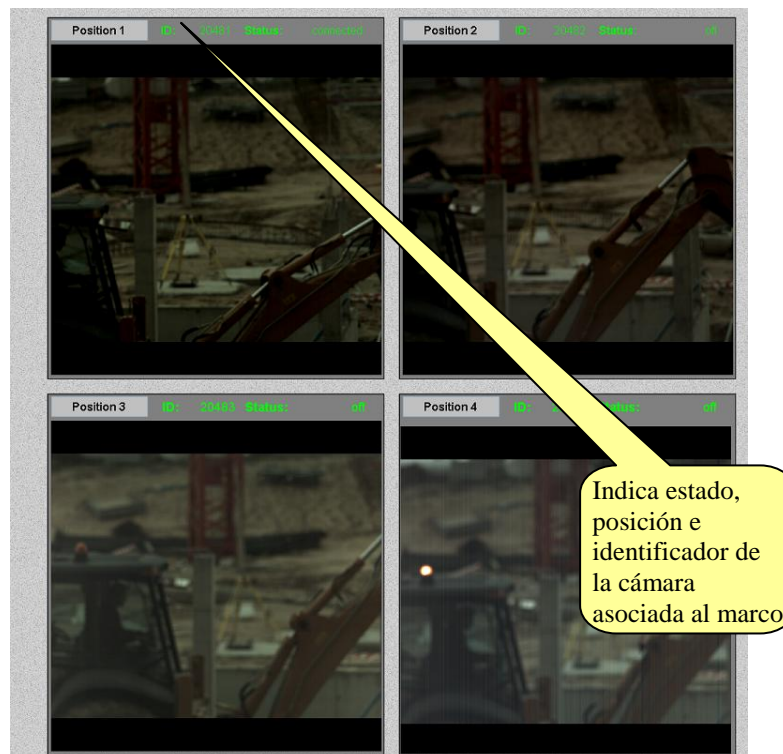
### A.3.1. Menú de Opciones



**Ilustración 53 : Detalle menú de opciones**

Situado en la parte superior de la ventana, permite realizar operaciones de guardado y recuperación del estado de la aplicación (parámetros de las cámaras, variables de alineamiento... ) y salir de la aplicación. Para que la recuperación de los parámetros sea posible, es necesario que haya el mismo número de cámaras conectadas que cuando se guardaron. Se incluyen unos iconos para con la mismas funcionalidad para usuarios acostumbrados a la interfaz

### A.3.2. Ventanas de visualización

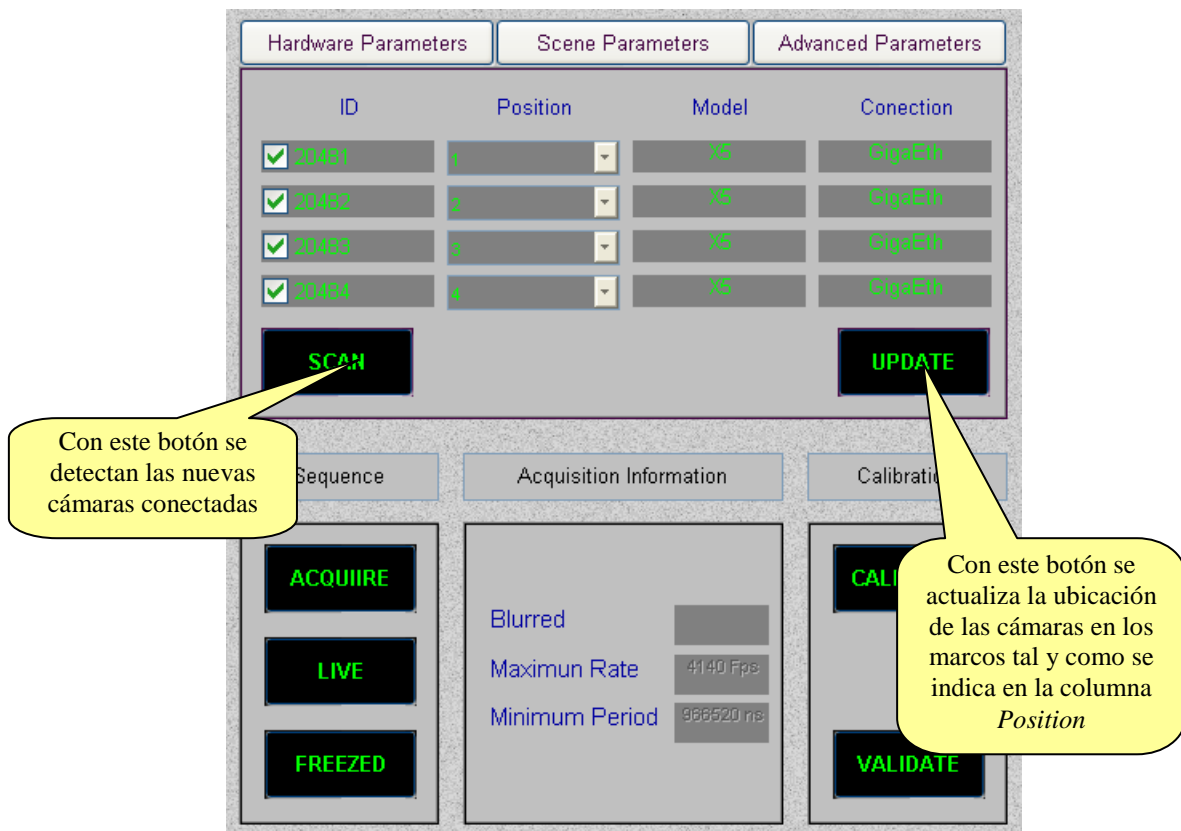


**Ilustración 54 : detalle de la interfaz de alineamiento y ajuste: marcos de imágenes**

En esta división, se visualizan imágenes de cada una de las cámaras de manera individual; además aparece cierta información acerca de las mismas: estado de la cámara (conectada, modo live, no habilitada...) e ID (identificador único para cada cámara). Ver Ilustración 54.

Es necesario, para el tratamiento post-captura de la secuencia, asociar las cámaras a los marcos acorde a la ubicación real de las mismas en la estructura, utilizando para ello el panel de control correspondiente, *Position* (ver Ilustración 55).

### A.3.3. Cuadro de control



**Ilustración 55 : detalle de la interfaz de alineamiento y ajuste: cuadros de control**

Con estos paneles se realiza el control y configuración del sistema: configuración de parámetros, colocación, visualización y alineamiento. Han sido desarrollados distintos paneles con el fin de clarificar y separar la funcionalidad de los mismos. A continuación se describen cada uno de los diferentes módulos:

#### ***HARDWARE PARAMETERS***

Pulsando sobre esta pestaña, situada en la parte superior del cuadro de control, aparece un panel (ver parte superior Ilustración 55) con los siguientes controles:



- Columna *ID*: identificador que se le asigna a cada cámara conectada, es único para cada una. El *stick* habilita/deshabilita la visualización en modo *live* de la cámara con el *ID* asociado en el marco correspondiente.
- Columna *Position*: Su valor indica en qué marco aparecerán las imágenes que capture la cámara con *ID* asociado. Según se ha indicado, estos desplegable deben configurarse para que los marcos muestren las imágenes en la misma disposición que se encuentran las cámaras.
- Columnas *Model* y *Conexión* : muestran información acerca del modelo ( X3 , X4 , Y3 ... ) y tipo de conexión ( USB 2.0 o *GigaEthernet*) de la cámara con el *ID* asociado
- Botón *SCAN*: rastrea los puertos USB y *Ethernet* de la maquina para detectar nuevas cámaras conectadas.
- Botón *UPDATE*: Hace efectiva o valida la configuración de los parámetros modificados, concretamente relaciona las cámaras con los marcos de las ventanas a través de los valores de la columna *Position*.

### SCENE PARAMETERS

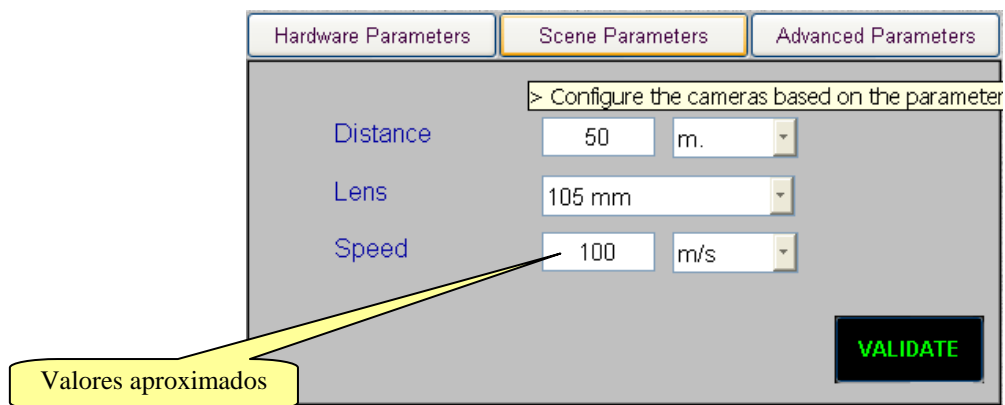


Ilustración 56 : detalle del interfaz de alineamiento y ajuste: Scene Parameters

Con este panel, ver Ilustración 56 , se pueden introducir parámetros relativos a la escena a grabar: velocidad del objeto, distancia de grabación y distancia focal (de la lente). Se supone que todas las cámaras utilizan lentes con la misma distancia focal ya que sino no sería posible alcanzar el objetivo. Esos parámetros son utilizados para calcular el valor de la estela borrosa, que se muestra en el panel *Acquisition Information* (Ilustración 55) y para tratamientos post – captura (la corrección radial depende de la lente utilizada).

### ADVANCED PARAMETERS



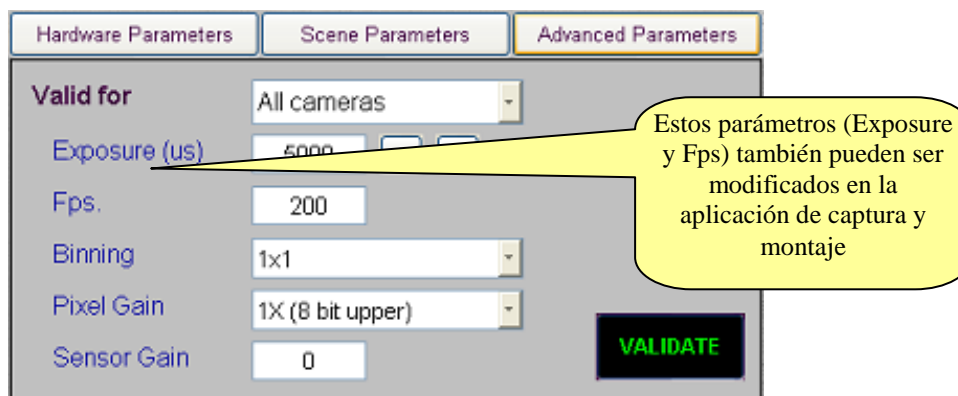


Ilustración 57 : detalle del interfaz de alineamiento y ajuste: Advanced Parameters

La última pestaña , Ilustración 57, permite configurar algunos parámetros de las cámaras, bien de manera individual o bien de manera conjunta. Las variables configurables son:

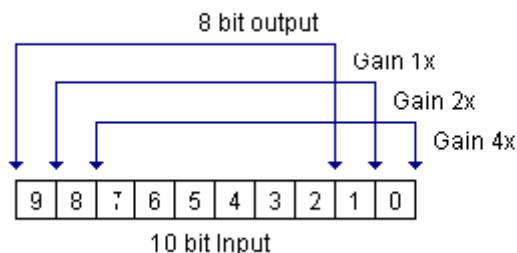
- **Binning:** este parámetro consiste en combinar píxeles adyacentes del sensor durante la captura. La principal ventaja de este parámetro es la mejora de la S/N de la imagen a costa de reducir la resolución espacial.

Tabla 11 : Binning , resolución y modelo de cámara

Binning	X-1 and X-3 (max size)	X-4 (max size)
1x1	1280 x 1024	512 x 512
2x2	640 x 512	256 x 256
3x3	424 x 340	168 x 168
4x4	320 x 256	128 x 128

- **Sensor Gain,** ganancia del sensor. El valor debe estar comprendido entre 0 (sin ganancia) y 3 (ganancia máxima). Es la cantidad de amplificación de la señal analógica. Se utiliza como un factor para maximizar rango dinámico de la cámara.
- **Píxel Gain,** ganancia digital: cada píxel es almacenado en la memoria con formato de 10 bits. El usuario sólo obtiene imágenes de 8 bits por píxel. Existen 3 maneras elegir el valor de estos pixeles que muestra la Tabla 12.

Tabla 12 : Ganancia de píxel



- *Exposure (ns)*: tiempo de exposición: tiempo que el obturador permanece abierto, de modo que el sensor recibe luz para la composición de la imagen.
- *Fps*, número de imágenes por segundo: medida de la tasa o ritmo con el que la cámara captura imágenes.

Es necesario especificar que pese a ser posible configurar las cámaras de manera individual, es recomendable que éstas sean configuradas de manera global. Aún obviando esta recomendación es absolutamente necesario que el tiempo de exposición sea único en el sistema para que la secuencia final compuesta sea coherente temporalmente.

## CALIBRATION

Con este bloque de botones, ver Ilustración 58 parte derecha, es posible realizar un alineamiento de las imágenes. Este alineamiento comienza pulsando el botón *CALIBRATE*; es entonces cuando comienza el proceso de alineamiento que consiste en que el usuario marca puntos homólogos sobre las imágenes de todas las cámaras; luego, pulsando el botón *VALIDATE*, se ejecuta un algoritmo que calcula las zonas visibles comunes de todas la imágenes captadas por las cámaras. Este algoritmo se detalla mas profundamente en la sección 4.4

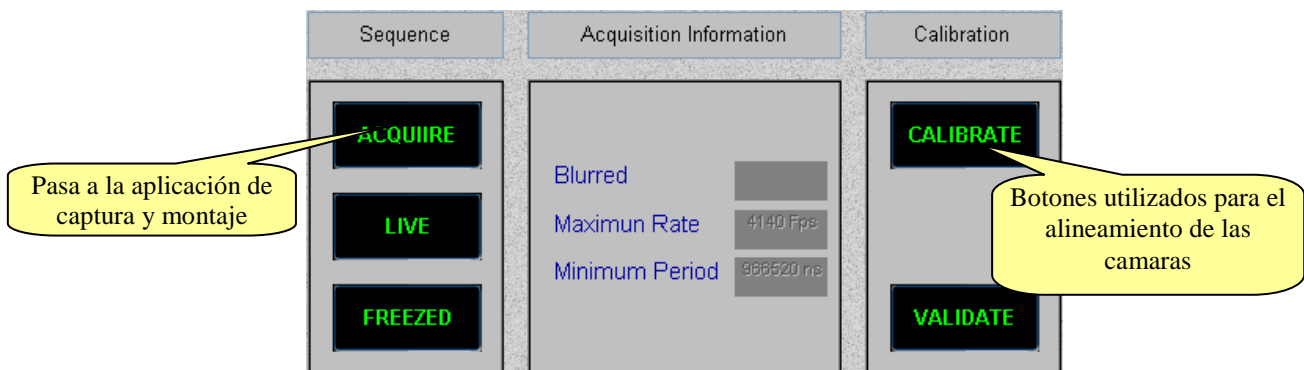


Ilustración 58 : detalle del interfaz de alineamiento y ajuste.  
Paneles de información, calibración y secuencia.

## SEQUENCE

En este panel (ver Ilustración 58 parte izquierda) se visualiza en los marcos de las ventanas en directo las imágenes capturadas por las cámaras pulsando el botón *LIVE*. En este modo no es posible configurar los parámetros de las cámaras; para salir, se pulsa el botón *FREZZE*, que *congela* la imagen captada en ese momento.

Una vez que todo está configurado, el botón *ACQUIRE* invoca a la pantalla de adquisición, captura y montaje de la secuencia, que se comenta en el apartado 3.3.3 .

## ACQUISITION INFORMATION

En la Ilustración 58 , el panel central aporta información acerca del sistema: estela borrosa, periodo máximo de grabación (número máximo de fps) y tiempo mínimo de exposición (en *ns*).

Estos datos, *Maximun Rate* y *Minimun Period*, se extraen directamente de las cámaras y dependen de la configuración de las mismas. El valor de la estela borrosa se calcula a partir de las variables relativas a la escena.

#### A.4. Modo AumentoTasa: Herramienta de captura y montaje

Esta herramienta (ver Ilustración 59) permite configurar las variables que determinan las características de la secuencia final, además una vez realizada la captura es posible descargar todas o una parte de las imágenes de la secuencia final a disco. En una futura mejora, esta herramienta incluirá también una interfaz para la generación de una secuencia mejorada utilizando las herramientas implementadas según el capítulo 4.

Una vez realizado el alineamiento y ajuste de parámetros, y antes de proceder a la grabación, es interesante poder previsualizar el aspecto que tendrá la secuencia resultante, sin tratamiento alguno, para poder confirmar que la etapa anterior esta bien realizada. Además esta herramienta ofrece la posibilidad de realizar un ajuste mas fino sobre las lentes de las cámaras (apertura del obturador y enfoque) para que las características de los sistemas ópticos utilizados sean lo mas parecidas posible. Ya que al visualizar la secuencia compuesta se acentúan las diferencias de contraste y perspectiva entre las imágenes de cada una de las cámaras.

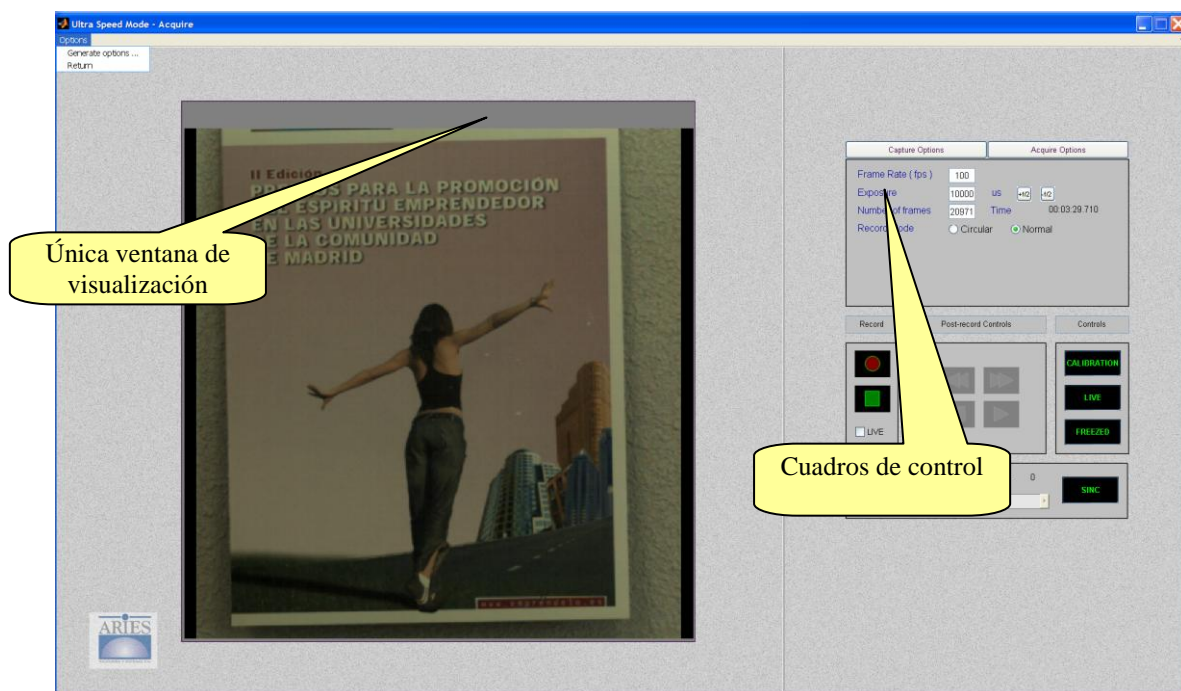


Ilustración 59 : Aspecto general del interfaz de captura y montaje

Después de este “ajuste” fino se puede proceder a la captura, pre-visualización y descarga a disco de parte o la totalidad de las imágenes de la secuencia capturada. Al igual que en la fase anterior, se dividió la pantalla en:

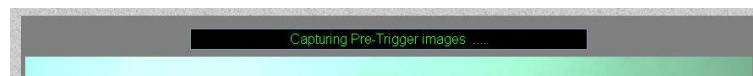
- Menú de opciones
- Ventana de visualización
- Cuadros de control

#### **A.4.1. Menú de opciones**

En este desplegable únicamente se permite volver a la aplicación de alineamiento y ajuste. En futuras mejoras, incluirá más opciones.

#### **A.4.2. Ventana de visualización**

Aparece un solo marco donde es posible visualizar la secuencia resultante de una etapa de captura del sistema. En este caso las imágenes no han sido tratadas y la única operación realizada sobre ella es la ordenación de las imágenes individuales de cada cámara. Se incluye un título, para mostrar información acerca del estado de la grabación u otra información necesarias, ver siguiente Ilustración



**Ilustración 60 : detalle de la interfaz de captura y montaje: marco.**

#### **A.4.3. Cuadros de control**

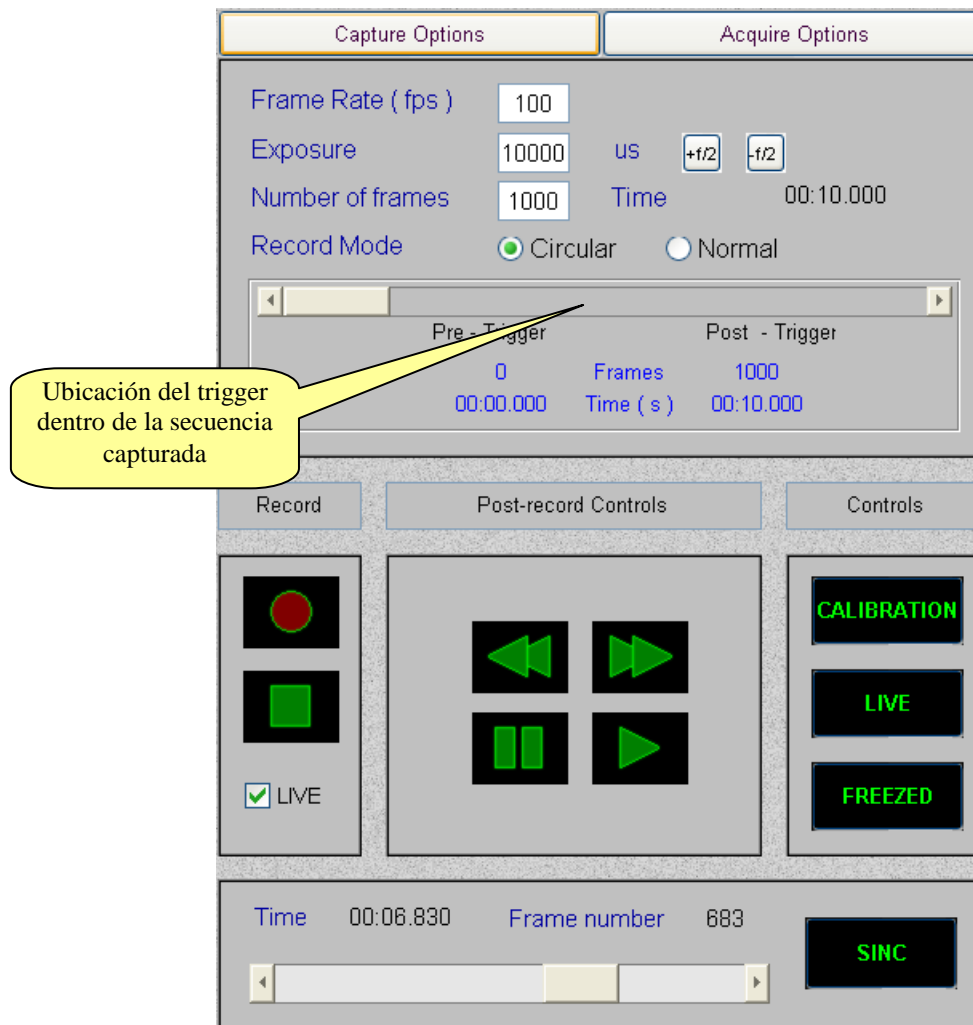


Ilustración 61: detalle de la interfaz de captura y montaje: cuadros de control

El orden en el que están descritos los cuadros de control coincide con el orden lógico de operación con el que se diseñaron para ser utilizados.

## ***CAPTURE OPTIONS***

Situado en la parte superior de este panel (Ver Ilustración 61) permite configurar distintos parámetros relacionados con las características de la secuencia a grabar como:

- *Frame Rate (fps)*: indica el número de imágenes por segundo que tendrá la secuencia resultante, esto implica que cada cámara capturar a una tasa igual a este valor dividido entre el número de cámaras del sistema. Es el mismo parámetro del panel de control *Advanced Parameters* de la aplicaron de ajuste y alineamiento.
- *Exposure*: tiempo de exposición de cada una de las cámaras. Este valor como mínimo será igual o menor que  $1/\text{Fps}$ . Es el mismo parámetro del panel de control *Advanced Parameters* de la aplicaron de ajuste y alineamiento.
- *Number of Frames*: número de imágenes que deseamos capturar. Con esta variable se calcula, junto con el fps, el tiempo total de grabación.

$$t_{\text{grabación}} = \text{fps} * \text{Number of Frame}$$

- *Record Mode* : indica el modo de grabación. Ante un *trigger* manual la aplicación puede operar de 3 maneras distintas. La primera que esta señal indique el comienzo de la grabación (modo *Normal*), la segunda que el *trigger* de por finalizada la grabación, esto implica que las cámaras capturan imágenes y descargan a la memoria de la cámara antes de que se produzca la señal (modo *Circular*). La tercera es una mezcla de las anteriores , es decir , que algunas imágenes se graben antes de *trigger* y el resto después (modo *Circular*).
  - Modo *Normal*: se pulsa el botón *rec* y la grabación comienza cuando se acciona el *trigger* manual. La grabación finaliza cuando termina el tiempo de grabación; para la cámara este tiempo finaliza cuando se llena toda la memoria que tiene asignada para la grabación.
  - Modo *Circular*: las cámaras capturan imágenes a la memoria interna de la cámara y reinician al comienzo de esta memoria cuando la memoria asignada (numero de imágenes y su resolución) esta totalmente llena. Las cámaras esperan un *trigger* para completar la adquisición . La posición del *slider* indica a la cámara el estado del *trigger* dentro de la grabación.

## **RECORD**

Este panel está localizado en la parte central izquierda del cuadros de control, Ilustración 61. Con el botón *REC* se prepara al sistema para grabar; configura algunos parámetros de las cámaras y el *timing hub*. La grabación no comenzará hasta que se accione de manera manual el *trigger* de las cámaras.

El botón *STOP* cancela la grabación pero no asegura que las imágenes capturadas sean “coherentes” temporalmente, este botón se utiliza para cancelar una grabación no para finalizarla. La grabación finaliza cuando se consume el tiempo de grabación y se produce una señal de *trigger*.

Con el *checkbox LIVE* se activa la visualización en directo mientras se realiza la grabación. Esta opción no es posible en todos los modelos de las cámaras.

## **POST-RECORD CONTROLS**

Una vez finalizada la grabación se puede visualizar la secuencia capturada utilizando estos controles ubicados en la parte central e inferior de los cuadros de control, Ilustración 61. Además, debajo de este panel se encuentra un *slider* para poder visualizar más fácilmente la grabación. Unos objetos tipo *text*, indican tiempo exacto de la grabación y número de imagen de la misma.

El botón *SINC* selecciona que cámara comenzó la grabación. Puesto que no podemos conocer a priori que cámara inicio la grabación, es necesario determinarlo de manera visual. Observemos este fenómeno con un ejemplo, para un ciclo de grabación con tres cámaras:

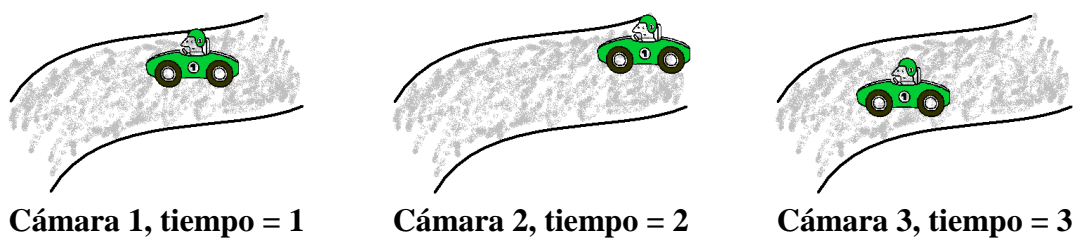


Ilustración 62 : *AumentoTasa* , ejemplo de ciclo de grabación para 3 cámaras (a)

Es necesario indicar que la cámara 2 comenzó la grabación, pulsando el botón SINC, el resultado deseado sería

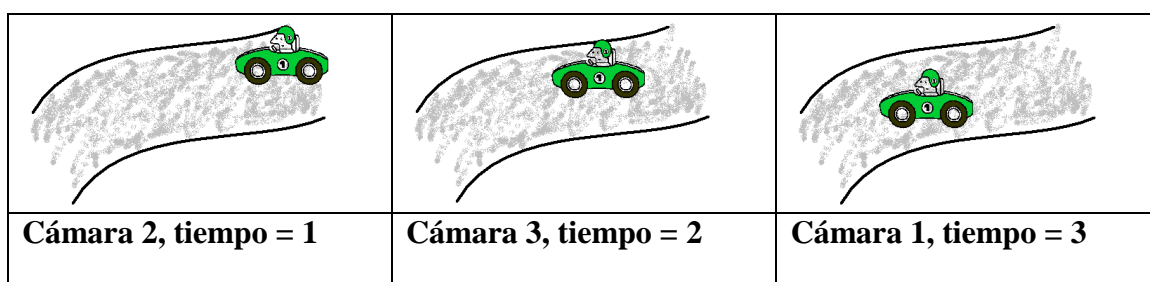


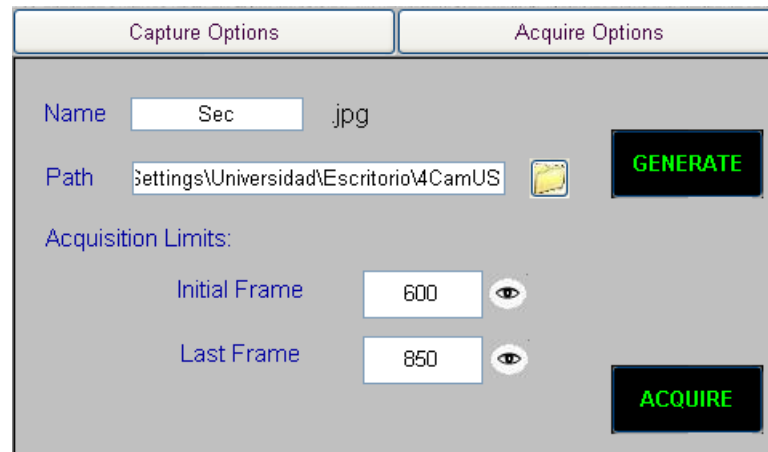
Ilustración 63 : *AumentoTasa* ejemplo de ciclo de grabación para 3 cámaras (b)

## PANEL CONTROLS

Este panel está localizado en la parte central derecha del cuadro de control , Ilustración 61. Pulsando el botón *CALIBRATION* volvemos a la ventana anterior, aplicación de ajuste y alineamiento, donde podemos configurar parámetros de las cámaras, guardar, cargar, salir de la aplicación...

Antes de realizar una captura es posible visualizar una simulación de la secuencia resultante final, pulsando sobre el botón *LIVE* se visualiza en cada instante una imagen sobre la ventana principal de cada una de las cámaras. Al igual que en la pantalla de *CALIBRATION*, durante la ejecución de este modo no es posible actualizar ningún parámetro de la cámara ni comenzar la grabación. Para finalizar el modo *LIVE* se pulsa sobre el botón *FREZZE*, que *congela* la visualización.

## CAPTURE OPTIONS



**Ilustración 64 : Detalle de la interfaz de captura y montaje: *Capture controls***

Con este panel se indica el nombre, tipo de formato, *path*, imagen inicial y final de la secuencia capturada que queremos descargar a disco. Téngase en cuenta que cada vez que se realiza una captura nueva (es decir, cada vez que se pulsa botón REC) se sobrescribe en la memoria interna de la cámara. Así que no es posible visualizar ni descargar a disco secuencias pertenecientes a otra fase de captura.

Utilizando los controles del cuadro *POST-RECORD* se puede visualizar la secuencia capturada y elegir qué parte se descargará al ordenador. En los *edit-box*, ver Ilustración 64, se puede indicar la imagen inicial (*Inicial Frame*) y final (*Last Frame*) que se desea descargar a disco. También la ubicación y el nombre de los archivos generados.

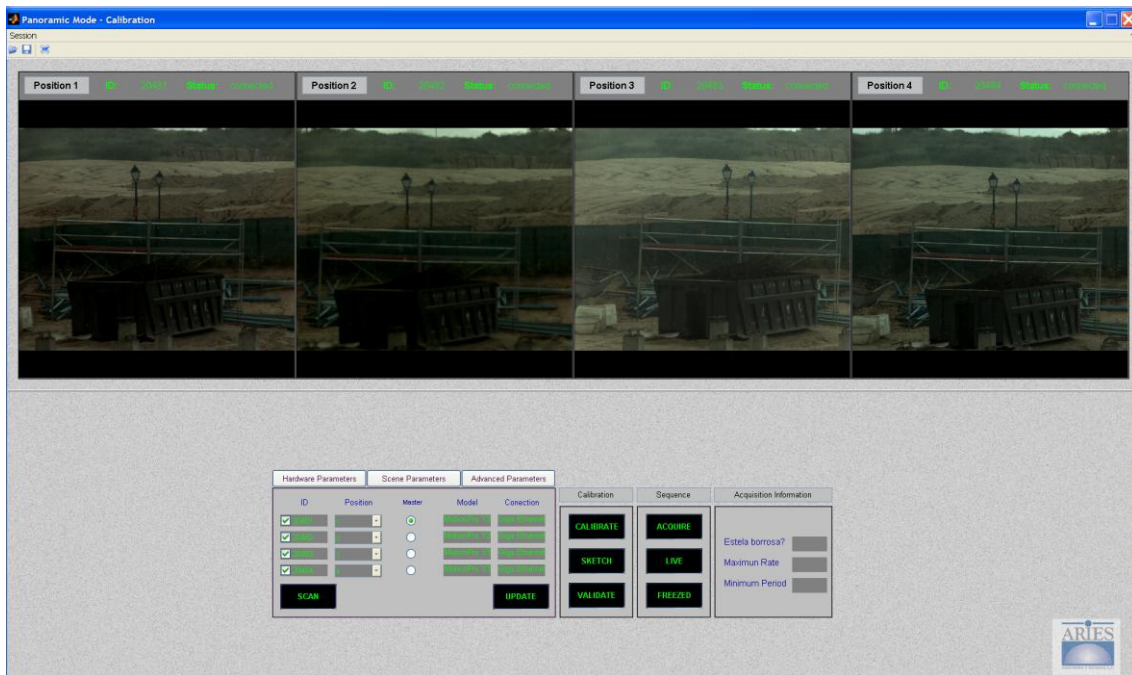
El botón de *Acquire* inicia la descarga según los parámetros introducidos en este panel. El botón *Generate* invocara una interfaz para configurar las correcciones a realizar sobre la secuencia final.

### **A.5. Modo Panorámico: Asistente de alineamiento y ajuste**

Esta aplicación ha sido desarrollada para configurar los parámetros del sistema de cámaras en el modo *AumentoCampo*. Hay que tener en cuenta que la mayoría de los controles desarrollados en esta aplicación son idénticos a los descritos en el apartado 3.3.2 . Por esto la funcionalidad de la aplicación es la misma salvo algunos detalles que se comentarán mas adelante.

Si se piensa en el sistema, ambos modos difieren en el sincronismo de la señal de captura. El resto de componentes (*trigger*, cámaras, conexiones...) es idéntico. Otra diferencia es la ubicación de las cámaras con respecto a la escena a capturar (no incumbe al *software* utilizado) y el asistente de alineamiento desarrollado para el modo *AumentoTasa*.





**Ilustración 65 :** Aspecto general del interfaz de alineamiento y ajuste para el modo *AumentoCampo*.

En la Ilustración 65, se observa el aspecto general de la interfaz para este modo. Si se compara con la Ilustración 12 , *AumentoTasa*, únicamente se modifican la ubicación de los objetos y pequeños detalles que a continuación se describen. De la misma manera queda separado el interfaz en tres regiones

- Menú de Opciones
- Ventanas de visualización
- Cuadro de controles

El menú de opciones, se describe en el apartado A.3.1. En cuanto a las ventanas de visualización se modifica su ubicación para simular la ubicación de las cámaras pero el contenido, información, es el mismo que en el descrito en el apartado A.3.2.

### ***CONTROLES COMUNES A AumentoTasa***

En este apartado se enumeran los controles que son comunes al modo *AumentoTasa*. La descripción de la funcionalidad de cada uno de ellos se encuentra en el apartado A.4.3.

- *Scene Parameters*
- *Advanced Parameters*
- *Acquisition Information*
- *Sequence*

## HARDWARE PARAMETERS

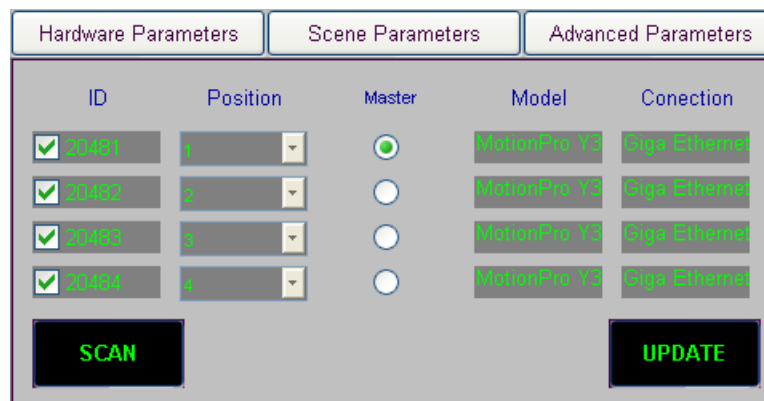


Ilustración 66 : Detalle interfaz alineamiento y ajuste para el modo *AumentoTasa* , *Hardware parameters*

Este panel de opciones se utiliza para ubicar las cámaras con sus marcos. Al igual que en el modo *AumentoTasa* , también se utiliza para detectar nuevas cámaras conectadas y aportar alguna información acerca del modelo de cámara y tipo de conexión.

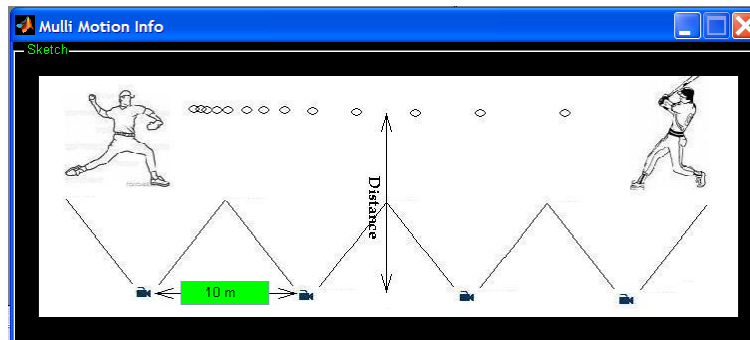
Este panel difiere del modo *AumentoTasa* en la columna *Master* , Ilustración 66 .Este conjunto de *radioButtons*, permite indicar que cámaras de las que están conectadas será denominada Master para el sistema. De este modo la cámara indicada como tal, en la etapa de captura, aporta la señal de sincronismo al resto de las cámaras conectadas para conseguir capturas simultaneas.

## CALIBRATION

La calibración en el modo *Panorámic*, no se trata de un alineamiento de las cámaras. Se comprobó que la alineación manual resultaba ser más efectiva que algunos métodos ideados.

El algoritmo de alineamiento, se intento buscar entre las zonas de solape alguna figura conocida, como una circunferencia o un folio DIN A3 . Y a partir de las regiones observadas en cada cámara hacer un recorte de las imágenes de las imágenes de cada cámara para componer la imagen panorámica.

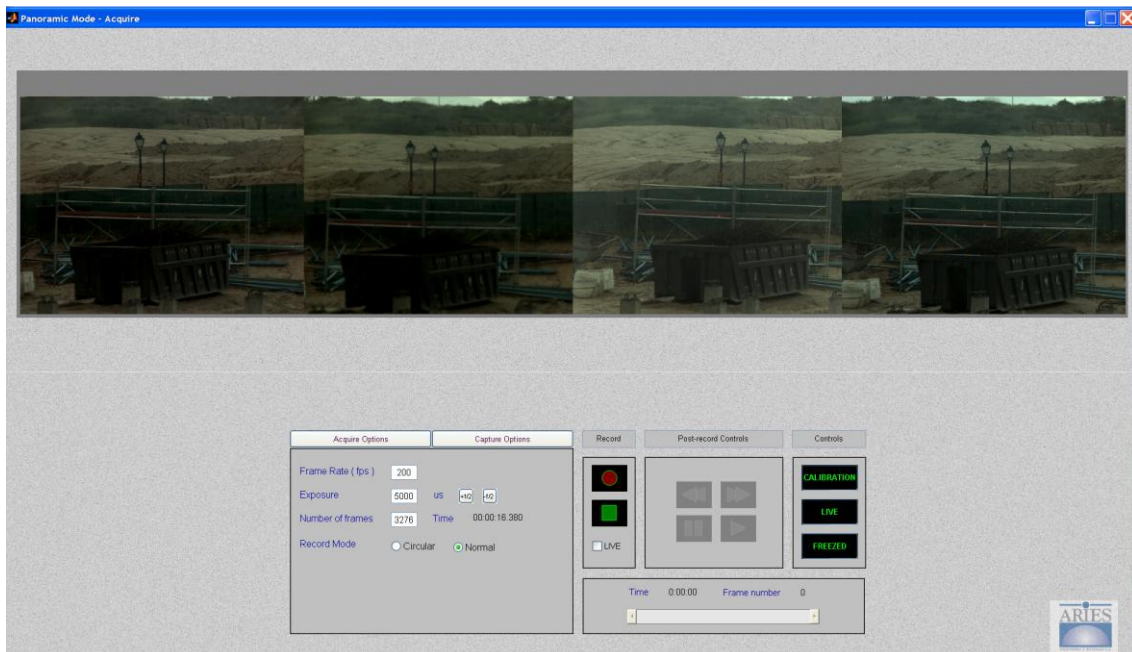
Aparte de los botones de *Calibration* y *Validate*, se encuentra en botón *sketch*. Este botón invoca a una ventana (Ilustración 67) donde se muestra un croquis con la separación necesaria entre las cámaras para conseguir una imagen panorámica. Estos valores depende de los parámetros relativos a la escena, *Scene Parameters* , por lo que si no han sido introducidos no aparecerá ninguna información.



**Ilustración 67 :** Ventana emergente cuando se pulsa en botón *sketch* que indica la separación necesaria para componer la panorámica utilizada en el modo *AumentoCampo*

### A.6. Modo *AumentoCampo*: Herramienta de captura y montaje

Esta herramienta, al igual que la desarrollada para el modo *AumentoTasa*, permite configurar las cámaras para realizar una captura. Los parámetros configurables en esta aplicación están relacionados con la secuencia final deseada como *fps* , *Exposure* , numero de imágenes, modo de grabación ... además de poder realizar una grabación, se puede visualizar parte de la secuencia capturada y descargarla a disco.



**Ilustración 68 :** *AumentoCampo* , interfaz de captura y montaje . Pan\_adquisicion.fig.

Aparte de las modificaciones realizadas en la colocación de los objetos, que se muestran en la Ilustración 15, en esta fase no se introdujeron nuevos objetos al diseño creado para el modo *AumentoTasa*.

Las modificaciones hechas en el código asociado para adaptarla al modo de grabación deseado se explicarán mas adelante en el apartado 3.4 , funcionamiento de la aplicación.

## A.7. Cinematic Analysis

Tal y como se expone en los requisitos iniciales, se ha desarrollado una herramienta para obtener estimaciones de los parámetros de movimiento del objeto filmado. Para seleccionar el objeto, el usuario selecciona marcando puntos móviles, puntos de giro y líneas de referencia sobre los objetos de la imagen. Esta aplicación es independiente de las anteriores y se utiliza *off-line* ya que el sistema de cámaras en esta etapa no es necesario.

Como punto de partida, esta aplicación necesita una secuencia de imágenes, numeradas, con un nombre común, y ubicadas en la misma carpeta. Esta secuencia es el resultado de las etapas de captura, montaje y descarga, para uno de los dos modos.

Para poder obtener mediciones aproximadas de la escena (obsérvese que se está trabajando con cámaras no calibradas), es necesario que un usuario marque referencias sobre algunas de las imágenes de la secuencia (mínimo 1 punto en al menos 3 imágenes). Además es necesario conocer el número de cuadros por segundo (fps) de la secuencia analizada y una relación píxel/longitud para poder ofrecer las medidas en unidades de la escena. La Ilustración 16 muestra el interfaz desarrollado. A continuación, se describe la interfaz y su funcionalidad

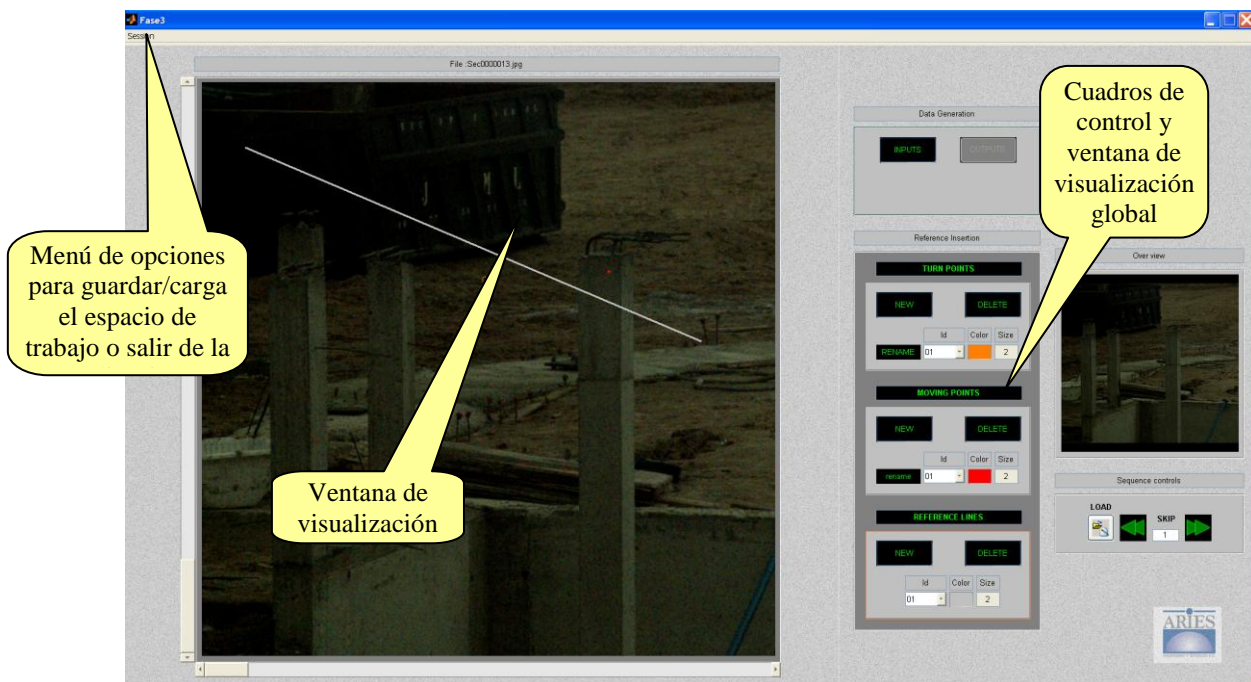


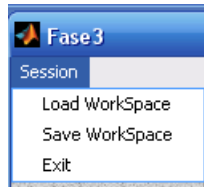
Ilustración 69 : Aspecto general de la interfaz de *Cinematic Analysis*

Esta aplicación, siguiendo un esquema similar a las presentadas hasta el momento, esta dividida en tres partes:

- Menú de Opciones

- Ventana de visualización
- Cuadros de control

### A.7.1. Menú de Opciones



**Ilustración 70: Menú de opciones de la interfaz de *Cinematic Analysis***

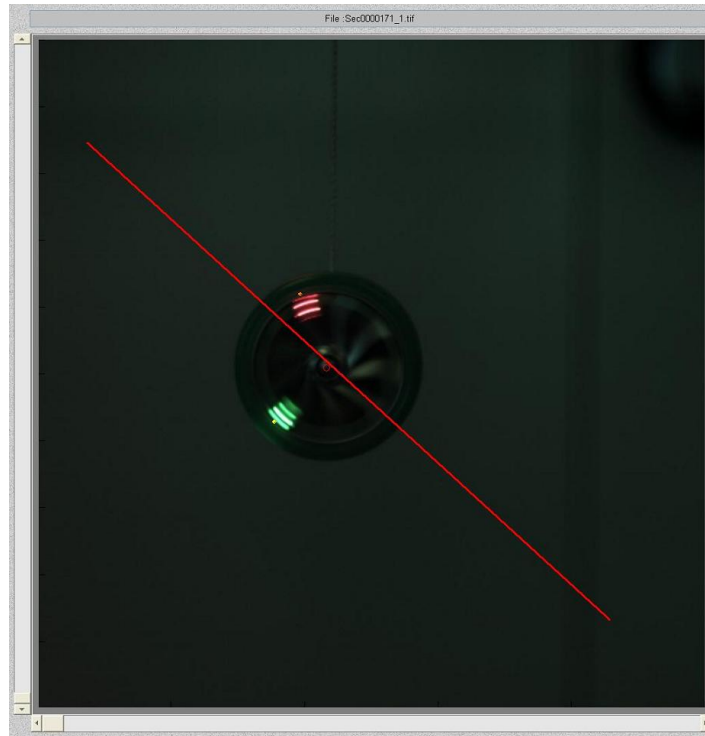
En la parte superior izquierda de la interfaz (Ilustración 69), está colocada la pestaña de *Session*, Ilustración 70. Pulsando sobre ella, aparece un menú despegable con las siguientes opciones:

- *Load WorkSpace* : permite cargar un espacio de trabajo grabado con anterioridad. Al seleccionarla aparece un explorador de sistema de archivos de donde escogeremos el archivo con extensión *.dat* , con el que queremos seguir trabajando. Este es el tipo de archivos que genera la esta aplicación al escoger la opción *Save*.
- *Save WorkSpace*: permite salvar el estado actual de la aplicación al disco; la extensión de este archivo será *.dat* y la ubicación la elige el usuario utilizando un explorador.
- *Exit*: finaliza la aplicación; antes pregunta al usuario si desea guardar el espacio de trabajo. También es posible finalizar la aplicación pulsando sobre el aspa situada en la parte superior derecha.

### A.7.2. Ventana de visualización

En este marco se visualiza una imagen de la secuencia cargada con una relación 1:1 (un píxel en el monitor representa un píxel de la imagen), con el fin de ofrecer máxima precisión en la visualización de las referencias marcadas. El nombre de la imagen se muestra en un cuadro en la parte superior, Ilustración 71. Superpuestos a la imagen además aparecerán objetos, puntos y líneas, que el usuario ha ubicado sobre ella. Estas referencias son utilizadas para realizar la estimación de los resultados. También aparecen *sliders* o barras de desplazamiento cuando la imagen que queremos visualizar es mayor, en píxeles, que las dimensiones del marco. En este caso la imagen se verá recortada, pudiéndola visualizar utilizando estas barras.





**Ilustración 71 : Detalle Interfaz de *Cinematic Analisis* , ventana de visualización**

### **A.7.3. Cuadros de control**

Este apartado esta compuesta por agrupaciones de botones para controlar las acciones que permiten la aplicación, y paneles para separar la funcionalidad de cada grupo de botones. A continuación se describe la funcionalidad de cada uno de los paneles que se visualizan en la interfaz , Ilustración 72, y los botones que lo componen.

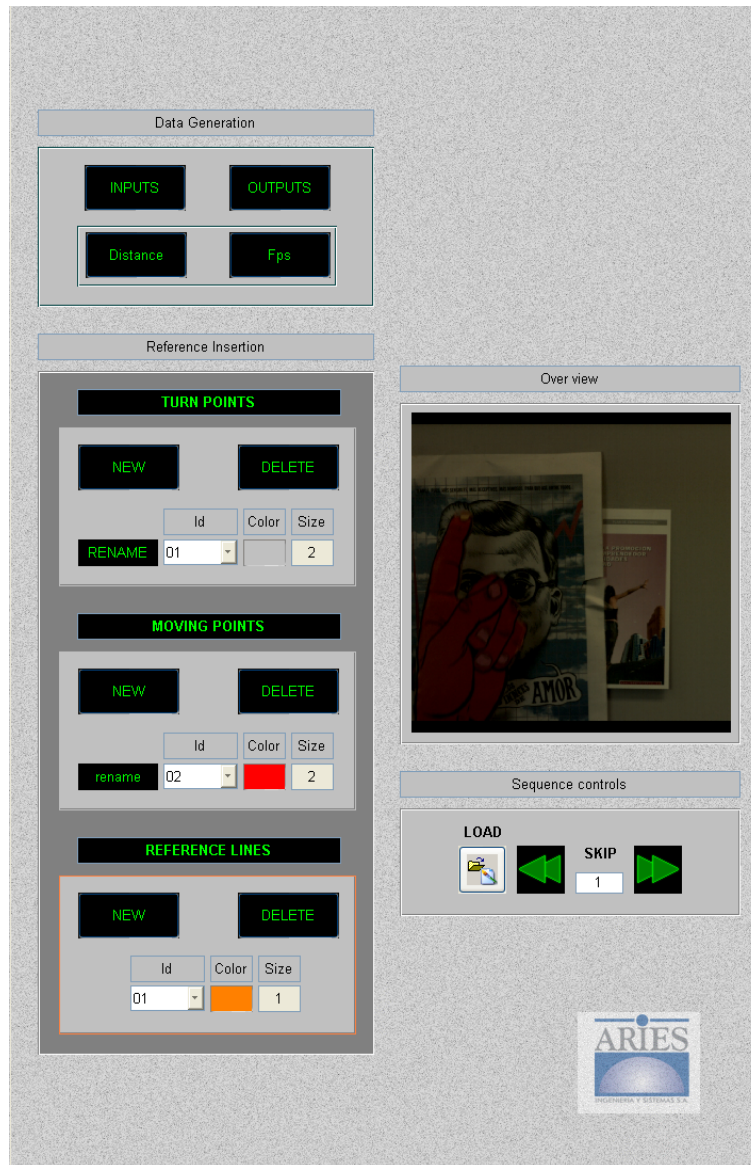






Ilustración 72 : Detalle Interfaz de *Cinematic Analysis* , cuadros de control.

## SEQUENCE CONTROLS

Controlan la visualización de las imágenes de la secuencia. Se carga una imagen pulsando sobre el botón , tras lo cual aparece un explorador y el usuario selecciona una de las imágenes de la secuencia que será la que se visualizará en la ventana principal. Para que un conjunto de archivos de imagen sea considerado como una secuencia, estos archivos deben tener las siguientes características:

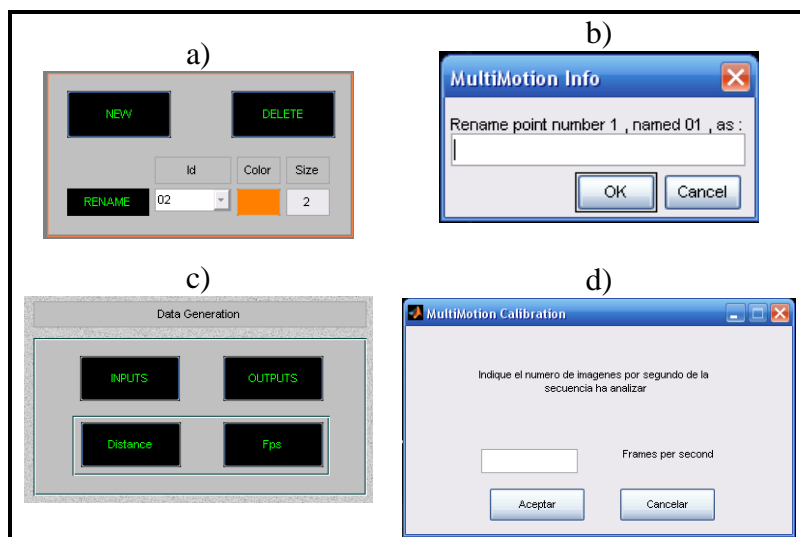
- NombreDeLosarchivosXXXXNombreXXXX.ext: el *Nombre* debe ser común a todos los archivos. *XXXX* indica el número de orden de la imagen en la secuencia numerada. Finalmente, *ext* se refiere a cualquier archivo de imagen que interprete MatLab.
- Todas las imágenes de la secuencia deben tener las mismas dimensiones.

A continuación, para visualizar el resto de las imágenes pertenecientes a la secuencia se pulsa el botón  para posteriores o bien  para anteriores. El número de imágenes que se avanza pulsando estos botones lo indica el campo .

## REFERENCE INSERTION

En este panel se gestionan los objetos o referencias marcados sobre las imágenes. Estos objetos pueden ser:

- *Turn Point*, o punto de giro: se tratan de puntos fijos de referencia para los puntos en movimiento. Un punto de giro marcado en una imagen es común para el resto de las imágenes: aparecerá en todas con la misma ubicación, salvo que se modifique o elimine. Sus coordenadas se utilizan para el cálculo de resultados.
- *Moving Point*, o punto en movimiento: se tratan del objeto clave para el cálculo de los resultados. Su ubicación es única en cada imagen.
- *Reference Line*, o línea de referencia: objeto común a todas las imágenes, igual que los puntos de giro. No se tienen en cuenta a la hora de realizar los cálculos, tan solo se utiliza para facilitar la ubicación de los puntos y puntos de giro.



**Ilustración 73 :** Detalles de la interfaz de *Cinematic Analisis*)Opciones con los objetos  
b)ventana emergente para modificar el nombre del objetoc)Opciones para la generacion de resultadosd)Ventana para la introducción del numero de fps de la secuencia analizada

Para los tres tipos de referencias, existe una manera única de operar (Ilustración 73 a ). Para crear una nueva referencia pulsamos sobre el botón *NEW* del panel correspondiente; esta acción inicia la creación de una nueva referencia, con *Id* consecutivo al ultimo creado (comenzando en  $Id = 1$ ).

A continuación se pulsa sobre la imagen para situar la referencia en ella (y así asociarle unas coordenadas). En caso de tratarse de un punto, ya sea móvil o de giro, al



pulsar sobre la imagen aparecerá una ventana con la zona ampliada para situarlo con una mayor precisión. Esta precisión es necesaria para la obtención de resultados coherentes.

Si se quiere reubicar una referencia en la imagen, seleccionamos el panel correspondiente (un borde anaranjado sobre el panel indicara que esta seleccionado ver Ilustración 73a) y utilizando el desplegable seleccionamos su *Id*. Ello permite modificar todos sus parámetros. Para eliminar una referencia de la imagen, se selecciona y se pulsa el botón *DELETE*. También es posible cambiar su tamaño en la imagen modificando el valor en la columna *Size* (siempre debe ser un numero entero positivo) y su *Color* pulsando sobre el cuadro del color del objeto. Aparecerá una paleta de colores donde se puede seleccionar el color.

Una opción especial, *RENAME* , permite modificar el campo *Id* de los puntos móviles y de los puntos de giro. Al pulsarlo aparece una ventana , Ilustración 73b , donde es posible introducir el nuevo nombre que se dará a la referencia; pulsando sobre el botón OK este cambio se hace efectivo. El *Id* de estos objetos se utiliza para nombrar la hojas de resultados del documento *MS Excel* que se crea como resultado del análisis; por eso resulta útil modificar este valor para identificar mas fácilmente a que objeto pertenecen los resultados.

## **DATA GENERATION**

Este panel, ver Ilustración 73c, genera los resultados a partir de los puntos marcados en las imágenes sobre la secuencia que se trabaja. Para estimar los resultados es necesario conocer dos variables acerca de la secuencia bajo estudio:

- Tasa de grabación de la secuencia: numero de imágenes por segundo de la secuencia analizada. Este valor se introduce a la aplicación pulsando sobre el botón *fps*. Aparece una ventana donde se introducen los valores que solicita la aplicación, ver Ilustración 73d
- Relación píxel/tamaño real, esta relación se obtiene conociendo una longitud real de un objeto en la imagen. La aplicación primero solicita que se marque sobre la imagen una línea recta, utilizando un *zoom* para conseguir más precisión. Después, una ventana solicita que se indique la longitud real de la línea que ha sido marcada. Con este procedimiento se obtiene la relación píxel-tamaño real (tanto más precisa cuanto más paralela a la cámara o plano de imagen sea la línea escogida, y cuanto más centrada se encuentre); se asume que los píxeles de las cámaras son cuadrados.

Una vez introducidos estos dos parámetros, es posible realizar el cálculo de los resultados. Para generar y exportar estos resultados se pulsa sobre el botón *OUTPUT*. Al hacerlo aparece un explorador donde el usuario debe indicar nombre y ubicación del archivo donde serán registrados los resultados.

Por conveniencia, estos resultados son generados en un formato legible por *MS Excel*, como hojas de libros, para así poder realizar con la máxima facilidad y flexibilidad operaciones de gestión y/o presentación de los resultados.

## ***OVERVIEW***

Este marco alberga una imagen cuadrada y completa del *frame* analizado. Un rectángulo amarillo indica qué parte de este *frame* está actualmente visible en la ventana de visualización. Con esta ventana el usuario puede visualizar la imagen completa que en el caso de ser muy grande no es posible visualizar en la ventana principal,

## PRESUPUESTO

- 1) **Ejecución Material**
  - Compra de ordenador personal (Software incluido)..... 6.000 €
  - Alquiler de cámaras durante 18 meses..... 3.600 €
  - Material de oficina ..... 150 €
  - Total de ejecución material ..... 9.750 €
  
- 2) **Gastos generales**
  - 16 % sobre Ejecución Material ..... 1560 €
  
- 3) **Beneficio Industrial**
  - 6 % sobre Ejecución Material ..... 586 €
  
- 4) **Honorarios Proyecto**
  - 1400 horas a 15 € / hora..... 21.000 €
  
- 5) **Material fungible**
  - Gastos de impresión..... 60 €
  - Encuadernación..... 200 €
  
- 6) **Subtotal del presupuesto**
  - Subtotal Presupuesto..... 33.156 €
  
- 7) **I.V.A. aplicable**
  - 16% Subtotal Presupuesto ..... 5.304 €
  
- 8) **Total presupuesto**
  - Total Presupuesto..... 38.460 €

Madrid, Julio de 2008

El Ingeniero Jefe de Proyecto

Fdo.: Eduardo Jiménez Sánchez  
Ingeniero de Telecomunicación

## PLIEGO DE CONDICIONES

Este documento contiene las condiciones legales que guiarán la realización, en este proyecto, de un HERRAMIENTAS DE APOYO A LA FUSIÓN DE SECUENCIAS DE VÍDEO DE ALTA VELOCIDAD PARA AUMENTAR LA RESOLUCIÓN TEMPORAL O EL CAMPO DE VISIÓN EFECTIVO. En lo que sigue, se supondrá que el proyecto ha sido encargado por una empresa cliente a una empresa consultora con la finalidad de realizar dicho sistema. Dicha empresa ha debido desarrollar una línea de investigación con objeto de elaborar el proyecto. Esta línea de investigación, junto con el posterior desarrollo de los programas está amparada por las condiciones particulares del siguiente pliego.

Supuesto que la utilización industrial de los métodos recogidos en el presente proyecto ha sido decidida por parte de la empresa cliente o de otras, la obra a realizar se regulará por las siguientes:

### Condiciones generales

1. La modalidad de contratación será el concurso. La adjudicación se hará, por tanto, a la proposición más favorable sin atender exclusivamente al valor económico, dependiendo de las mayores garantías ofrecidas. La empresa que somete el proyecto a concurso se reserva el derecho a declararlo desierto.

2. El montaje y mecanización completa de los equipos que intervengan será realizado totalmente por la empresa licitadora.

3. En la oferta, se hará constar el precio total por el que se compromete a realizar la obra y el tanto por ciento de baja que supone este precio en relación con un importe límite si este se hubiera fijado.

4. La obra se realizará bajo la dirección técnica de un Ingeniero Superior de Telecomunicación, auxiliado por el número de Ingenieros Técnicos y Programadores que se estime preciso para el desarrollo de la misma.

5. Aparte del Ingeniero Director, el contratista tendrá derecho a contratar al resto del personal, pudiendo ceder esta prerrogativa a favor del Ingeniero Director, quien no estará obligado a aceptarla.

6. El contratista tiene derecho a sacar copias a su costa de los planos, pliego de condiciones y presupuestos. El Ingeniero autor del proyecto autorizará con su firma las copias solicitadas por el contratista después de confrontarlas.

7. Se abonará al contratista la obra que realmente ejecute con sujeción al proyecto que sirvió de base para la contratación, a las modificaciones autorizadas por la superioridad o a las órdenes que con arreglo a sus facultades le hayan comunicado por escrito al Ingeniero Director de obras siempre que dicha obra se haya ajustado a los preceptos de los pliegos de condiciones, con arreglo a los cuales, se harán las modificaciones y la valoración de las diversas unidades sin que el importe total pueda exceder de los presupuestos aprobados. Por consiguiente, el número de unidades que se consignan en el proyecto o en el presupuesto, no podrá servirle de fundamento para entablar reclamaciones de ninguna clase, salvo en los casos de rescisión.

8. Tanto en las certificaciones de obras como en la liquidación final, se abonarán los trabajos realizados por el contratista a los precios de ejecución material que figuran en el presupuesto para cada unidad de la obra.

9. Si excepcionalmente se hubiera ejecutado algún trabajo que no se ajustase a las condiciones de la contrata pero que sin embargo es admisible a juicio del Ingeniero Director de obras, se dará conocimiento a la Dirección, proponiendo a la vez la rebaja de precios que el Ingeniero estime justa y si la Dirección resolviera aceptar la obra, quedará el contratista obligado a conformarse con la rebaja acordada.

10. Cuando se juzgue necesario emplear materiales o ejecutar obras que no figuren en el presupuesto de la contrata, se evaluará su importe a los precios asignados a otras obras o materiales análogos si los hubiere y cuando no, se discutirán entre el Ingeniero Director y el contratista, sometiéndolos a la aprobación de la Dirección. Los nuevos precios convenidos por uno u otro procedimiento, se sujetarán siempre al establecido en el punto anterior.

11. Cuando el contratista, con autorización del Ingeniero Director de obras, emplee materiales de calidad más elevada o de mayores dimensiones de lo estipulado en el proyecto, o sustituya una clase de fabricación por otra que tenga asignado mayor precio o ejecute con mayores dimensiones cualquier otra parte de las obras, o en general, introduzca en ellas cualquier modificación que sea beneficiosa a juicio del Ingeniero Director de obras, no tendrá derecho sin embargo, sino a lo que le correspondería si hubiera realizado la obra con estricta sujeción a lo proyectado y contratado.

12. Las cantidades calculadas para obras accesorias, aunque figuren por partida alzada en el presupuesto final (general), no serán abonadas sino a los precios de la contrata, según las condiciones de la misma y los proyectos particulares que para ellas se formen, o en su defecto, por lo que resulte de su medición final.

13. El contratista queda obligado a abonar al Ingeniero autor del proyecto y director de obras así como a los Ingenieros Técnicos, el importe de sus respectivos honorarios facultativos por formación del proyecto, dirección técnica y administración en su caso, con arreglo a las tarifas y honorarios vigentes.

14. Concluida la ejecución de la obra, será reconocida por el Ingeniero Director que a tal efecto designe la empresa.

15. La garantía definitiva será del 4% del presupuesto y la provisional del 2%.

16. La forma de pago será por certificaciones mensuales de la obra ejecutada, de acuerdo con los precios del presupuesto, deducida la baja si la hubiera.

17. La fecha de comienzo de las obras será a partir de los 15 días naturales del replanteo oficial de las mismas y la definitiva, al año de haber ejecutado la provisional, procediéndose si no existe reclamación alguna, a la reclamación de la fianza.

18. Si el contratista al efectuar el replanteo, observase algún error en el proyecto, deberá comunicarlo en el plazo de quince días al Ingeniero Director de obras, pues transcurrido ese plazo será responsable de la exactitud del proyecto.

19. El contratista está obligado a designar una persona responsable que se entenderá con el Ingeniero Director de obras, o con el delegado que éste designe, para todo relacionado con ella. Al ser el Ingeniero Director de obras el que interpreta el proyecto, el contratista deberá consultarle cualquier duda que surja en su realización.

20. Durante la realización de la obra, se girarán visitas de inspección por personal facultativo de la empresa cliente, para hacer las comprobaciones que se crean oportunas. Es obligación del contratista, la conservación de la obra ya ejecutada hasta la recepción de la

misma, por lo que el deterioro parcial o total de ella, aunque sea por agentes atmosféricos u otras causas, deberá ser reparado o reconstruido por su cuenta.

21. El contratista, deberá realizar la obra en el plazo mencionado a partir de la fecha del contrato, incurriendo en multa, por retraso de la ejecución siempre que éste no sea debido a causas de fuerza mayor. A la terminación de la obra, se hará una recepción provisional previo reconocimiento y examen por la dirección técnica, el depositario de efectos, el interventor y el jefe de servicio o un representante, estampando su conformidad el contratista.

22. Hecha la recepción provisional, se certificará al contratista el resto de la obra, reservándose la administración el importe de los gastos de conservación de la misma hasta su recepción definitiva y la fianza durante el tiempo señalado como plazo de garantía. La recepción definitiva se hará en las mismas condiciones que la provisional, extendiéndose el acta correspondiente. El Director Técnico propondrá a la Junta Económica la devolución de la fianza al contratista de acuerdo con las condiciones económicas legales establecidas.

23. Las tarifas para la determinación de honorarios, reguladas por orden de la Presidencia del Gobierno el 19 de Octubre de 1961, se aplicarán sobre el denominado en la actualidad "Presupuesto de Ejecución de Contrata" y anteriormente llamado "Presupuesto de Ejecución Material" que hoy designa otro concepto.

### **Condiciones particulares**

La empresa consultora, que ha desarrollado el presente proyecto, lo entregará a la empresa cliente bajo las condiciones generales ya formuladas, debiendo añadirse las siguientes condiciones particulares:

1. La propiedad intelectual de los procesos descritos y analizados en el presente trabajo, pertenece por entero a la empresa consultora representada por el Ingeniero Director del Proyecto.

2. La empresa consultora se reserva el derecho a la utilización total o parcial de los resultados de la investigación realizada para desarrollar el siguiente proyecto, bien para su publicación o bien para su uso en trabajos o proyectos posteriores, para la misma empresa cliente o para otra.

3. Cualquier tipo de reproducción aparte de las reseñadas en las condiciones generales, bien sea para uso particular de la empresa cliente, o para cualquier otra aplicación, contará con autorización expresa y por escrito del Ingeniero Director del Proyecto, que actuará en representación de la empresa consultora.

4. En la autorización se ha de hacer constar la aplicación a que se destinan sus reproducciones así como su cantidad.

5. En todas las reproducciones se indicará su procedencia, explicitando el nombre del proyecto, nombre del Ingeniero Director y de la empresa consultora.

6. Si el proyecto pasa la etapa de desarrollo, cualquier modificación que se realice sobre él, deberá ser notificada al Ingeniero Director del Proyecto y a criterio de éste, la empresa consultora decidirá aceptar o no la modificación propuesta.

7. Si la modificación se acepta, la empresa consultora se hará responsable al mismo nivel que el proyecto inicial del que resulta el añadirla.

8. Si la modificación no es aceptada, por el contrario, la empresa consultora declinará toda responsabilidad que se derive de la aplicación o influencia de la misma.

9. Si la empresa cliente decide desarrollar industrialmente uno o varios productos en los que resulte parcial o totalmente aplicable el estudio de este proyecto, deberá comunicarlo a la empresa consultora.

10. La empresa consultora no se responsabiliza de los efectos laterales que se puedan producir en el momento en que se utilice la herramienta objeto del presente proyecto para la realización de otras aplicaciones.

11. La empresa consultora tendrá prioridad respecto a otras en la elaboración de los proyectos auxiliares que fuese necesario desarrollar para dicha aplicación industrial, siempre que no haga explícita renuncia a este hecho. En este caso, deberá autorizar expresamente los proyectos presentados por otros.

12. El Ingeniero Director del presente proyecto, será el responsable de la dirección de la aplicación industrial siempre que la empresa consultora lo estime oportuno. En caso contrario, la persona designada deberá contar con la autorización del mismo, quien delegará en él las responsabilidades que ostente.