

Máster Universitario en Ingeniería Industrial

2015 – 2017

Trabajo Fin de Máster

“Aplicaciones Técnicas de Visión Artificial a Balística de Efectos”

Álvaro S. Fernández Malagón

Tutor

José Antonio Loya Lorenzo

Dpto. Mecánica de Medios Continuos y Teoría de Estructura

Leganés 14 de julio de 2017

Palabras clave: Matlab, Visión Artificial, Balística de Efectos.

Resumen: Se han llevado a cabo 6 ensayos con 3 clases distintas de proyectil los cuales han sido grabados con cámaras de alta velocidad con el objetivo de poder desarrollar una aplicación en Matlab que permita ver cómo actúan los proyectiles en el interior del cuerpo humano (gelatina balística) y de esta manera poder conocer el daño que provocan en tiempo real. Esta aplicación creada en Matlab es capaz de calcular la posición, la velocidad, la aceleración y la energía transmitidas por el proyectil y el área y volumen generados por este en el cuerpo. Estos datos posteriormente son grabados en un Excel.



Esta obra se encuentra sujeta a la licencia Creative Commons **Reconocimiento – No Comercial – Sin Obra Derivada**

Me gustaría dar las gracias a mi tutor José Antonio Loya Lorenzo por su paciencia y dedicación.

A mi madre, Conchi, por ese apoyo que siempre me ha dado incluso en los momentos más difíciles.

A mi padre, porque siempre me ha ayudado y ha confiado en mí.

Y a mi abuela, que este donde este, continúa protegiéndome y cuidándome.

Índice

1	INTRODUCCIÓN, OBJETIVOS Y ESTRUCTURA	6
1.1	INTRODUCCIÓN	6
1.2	OBJETIVOS	7
1.3	ESTRUCTURA DE LA MEMORIA.....	7
2	BÁLISTICA TERMINAL O DE EFECTOS	9
2.1	¿QUÉ ES?	9
2.2	FACTORES QUE INFLUYEN EN LA BALÍSTICA TERMINAL	10
2.2.1	Velocidad del proyectil	10
2.2.2	Ángulo de incidencia.....	11
2.2.3	Características del proyectil.....	11
2.2.4	Distancia de disparo	12
2.3	CAVITACIÓN TEMPORAL DEL PROYECTIL.....	14
3	VISIÓN ARTIFICIAL	17
3.1	¿CÓMO FUNCIONA?.....	17
3.2	OBJETIVOS	18
3.3	PROCESAMIENTO DIGITAL DE LAS IMÁGENES Y ETAPAS QUE LO COMPONEN	18
3.4	ELEMENTOS	20
3.4.1	Iluminación del sistema de visión artificial.....	20
3.4.2	Ópticas o lentes	21
3.4.3	Cámara: definición y características.....	21
3.4.4	Programación	21
3.4.5	Hardware: Framegrabbers.....	22
3.4.6	Softwares empleados en la visión artificial	23
3.5	APLICACIONES	23
4	MATLAB Y PROCESADO DIGITAL DE IMÁGENES.....	25
4.1	¿QUÉ ES MATLAB?	25
4.2	INTRODUCCIÓN AL PROCESADO DIGITAL DE IMAGENES.....	26
4.2.1	Definición.....	26
4.2.2	Historia	27
4.2.3	Áreas de aplicación	28
4.2.4	Clases principales de procesado digital de imágenes	33
4.2.5	Pasos fundamentales en el procesado digital de imágenes	34
4.3	INTRODUCCIÓN AL PROCESADO DIGITAL DE IMÁGENES CON MATLAB.....	35
4.3.1	El entorno de Matlab.....	36
4.3.2	El escritorio de Matlab	36
4.3.3	Variables y ficheros.....	37
4.3.4	Comandos y matrices	38
4.3.5	Representación de imágenes.....	41
4.3.6	Entrada y salida de imágenes	42
4.3.7	Tipos de datos y de imágenes.....	43
4.4	IMAGE PROCESSING TOOLBOX.....	46
5	ENSAYOS REALIZADOS	48
5.1	PROYECTILES.....	48
5.1.1	FULL METAL JACKET (FMJ).....	48
5.1.2	PUNTA BLANDA (SP)	49

Aplicaciones Técnicas de Visión Artificial a Balística de Efectos

5.1.3	PUNTA HUECA (HP)	50
5.2	GELATINA BALÍSTICA.....	51
5.3	CÁMARA	52
5.4	ILUMINACIÓN	52
6	MANUAL DE FUNCIONAMIENTO	53
6.1	PHOTRON FASTCAM VIEWER	53
6.2	MATLAB.....	58
7	PLANIFICACIÓN.....	70
8	PRESUPUESTO	71
8.1	PARTIDA DE PERSONAL.....	71
8.2	PARTIDA DE MATERIALES.....	72
8.3	RESUMEN DE COSTES.....	72
9	CONCLUSIONES Y LÍNEAS FUTURAS.....	74
9.1	CONCLUSIONES.....	74
9.2	LÍNEAS FUTURAS.....	74
10	BIBLIOGRAFÍA	76
10.1	LIBROS DE CONSULTA	76
10.2	PÁGINAS WEB	76
11	ANEXOS	78
11.1	ANEXO I: CÓDIGO	78
11.2	ANEXO II: RESULTADOS PROYECTIL SP	86
11.3	ANEXO III: RESULTADOS PROYECTIL HP	87

Índice de Figuras

Figura 2-1: Lesiones producidas por los proyectiles de alta velocidad (arriba) y los de baja velocidad (abajo).....	10
Figura 2-2: Ángulo de incidencia de un proyectil en un cuerpo.....	11
Figura 2-3: Efecto plugging, debido a balas tipo romo.....	12
Figura 2-4: Distancia de disparo	14
Figura 2-5: Cavitación temporal producida por un proyectil [2].....	14
Figura 2-6: Proyectil fmj impactando con la gelatina balística.	15
Figura 2-7: Proyectil fmj penetrando en la gelatina balística.	15
Figura 2-8: Proyectil fmj avanza por el interior de la gelatina balística.	16
Figura 2-9: Salida proyectil fmj de la gelatina balística.	16
Figura 2-10: Crecimiento del orificio generado por el proyectil debido a la cavitación temporal	16
Figura 3-1: Visión artificial frente a la visión humana.	18
Figura 3-2: Etapas del procesamiento digital de las imágenes.	19
Figura 3-3: Imagen analógica frente a imagen digitalizada.....	19
Figura 3-4: Importancia de iluminación a la hora de resaltar defectos superficiales.....	20
Figura 3-5: Proceso operativo del sistema.	22
Figura 3-6:Etapas del proceso de captura de imágenes.	22
Figura 3-7: Aplicaciones visión artificial [1].....	24
Figura 4-1: Espectro electromagnético [7].	29
Figura 4-2: Escáner PET del cerebro humano.	29
Figura 4-3: Cerebro de un ratón con luz ultravioleta.....	30
Figura 4-4: Detección de vegetación con imagen infrarroja.....	31
Figura 4-5: Montañas en el Tíbet vistas por un radar topográfico.....	32
Figura 4-6: Detección de emisiones de gases en el fondo del Golfo de México con sonar....	33
Figura 4-7: Circuito integrado ampliado 2000 veces con SEM.....	33
Figura 4-8: Interface de Matlab.	36
Figura 4-9: Matriz en Matlab.....	39
Figura 5-1: Proyectil Full Metal Jacket.	48
Figura 5-2: Proyectil Punta Blanda.	49
Figura 5-3: Proyectil Punta Hueca.	50
Figura 5-4: Gelatina Balística [24].	51
Figura 5-5: Cámara Photron Fastcam SA-Z	52
Figura 5-6: Iluminación.	52
Figura 6-1: Interface Photron Fastcam Viewer.....	53
Figura 6-2: Selección del archivo.	54
Figura 6-3: Interface del programa con el archivo abierto.	55
Figura 6-4: Información aportada por el archivo *.cih.....	55
Figura 6-5: Introducción frames penetración y fin de expansión.....	56
Figura 6-6: Guardas los frames.....	56
Figura 6-7: Selección de calidad de las imágenes.	57
Figura 6-8: Imágenes en la carpeta original.....	57
Figura 6-9: Interface Matlab	58
Figura 6-10: Guardas los frames.....	59
Figura 6-11: Llamada al programa desde el Command Window	59
Figura 6-12: Ventana para localizar carpeta.	60
Figura 6-13: Indicar lado.	60
Figura 6-14: Determinar la masa de la bala.....	60
Figura 6-15: Determinar el tamaño de la gelatina balística.	60

Aplicaciones Técnicas de Visión Artificial a Balística de Efectos

Figura 6-16: Determinar los frames por segundo.....	61
Figura 6-17: Ventana emergente para realizar primer corte.....	61
Figura 6-18: Cortar Imagen.....	61
Figura 6-19: Indicar si se quiere realizar de nuevo el corte.....	62
Figura 6-20: Definir umbral.....	62
Figura 6-21: Binarizado imagen.....	62
Figura 6-22: Ventana emergente para realizar segundo corte.....	63
Figura 6-23: Segundo corte.....	63
Figura 6-24: Indicar si se quiere realizar de nuevo el corte.....	64
Figura 6-25: Video en el que se observa el avance del proyectil.....	64
Figura 6-26: Tabla con información sobre el proyectil.....	65
Figura 6-27: Error en el cálculo de la fisura.....	65
Figura 6-28: Confirmación error.....	66
Figura 6-29: Definir imagen a partir de la cual se produce el error.....	66
Figura 6-30: Video en el que se observa el avance del proyectil.....	66
Figura 6-31: Información proyectil y orificio.....	67
Figura 6-32: Cantidad de energía intercambiada.....	67
Figura 6-33: Gráfica posición frente al tiempo.....	68
Figura 6-34: Gráfica de la velocidad frente al tiempo.....	68
Figura 6-35: Gráfica de la aceleración frente al tiempo.....	69
Figura 7-1: Diagrama de Gantt.....	70
Figura 11-1: Posición frente al tiempo.....	86
Figura 11-2: Velocidad frente al tiempo.....	86
Figura 11-3: Aceleración frente al tiempo.....	87
Figura 11-4: Posición frente al tiempo.....	87
Figura 11-5: Velocidad frente al tiempo.....	88
Figura 11-6: Aceleración frente al tiempo.....	88

Índice de Tablas

Tabla 1: Tipos de datos con los que trabaja Matlab.....	44
Tabla 2: Funciones para la conversión de datos.	44
Tabla 3: Funciones para la conversión de imágenes.....	46
Tabla 4: Planificación de tareas.	70
Tabla 5: Días dedicados por cada ingeniero a cada tarea.	71
Tabla 6: Coste de personal.....	71
Tabla 7: Costes Materiales.....	72
Tabla 8: Coste total proyecto.	73

1 INTRODUCCIÓN, OBJETIVOS Y ESTRUCTURA

1.1 INTRODUCCIÓN

La visión artificial o visión por computador (computer vision) pertenece al campo de la inteligencia artificial. Consiste en programar un ordenar de tal manera que sea capaz de interpretar las características de una imagen o entender una situación. En definitiva, tratar de actuar como el ojo humano, el cual es capaz de captar un objeto mediante los receptores de la retina (cámara) y transformarlos en impulsos nerviosos que son enviados al cerebro (ordenador) y procesados por este.

Resulta de máximo interés para las fuerzas y cuerpos de seguridad del Estado conocer los efectos que provocan los distintos tipos de municiones de armas de fuego al incidir sus proyectiles sobre el cuerpo humano y otro tipo de superficies. Por este motivo, es habitual realizar disparos de arma de fuego contra bloques de gelatina balística, ya que en estas condiciones, presenta un comportamiento semejante al de la carne humana.

Dadas las características de este tipo de ensayos, tras el impacto, se suelen analizar algunos parámetros representativos mediante inspección visual, como la penetración máxima alcanzada, la trayectoria seguida por el proyectil o la cavidad permanente generada, entre otros. Sin embargo, es de gran interés obtener información también durante todo el proceso de impacto y perforación.

Dadas las características de transparencia de la gelatina, en este trabajo se propone aplicar técnicas de fotografía de alta velocidad que permitan filmar todo el proceso de impacto y penetración del proyectil y así obtener información adicional en cada instante de tiempo.

Se han llevado a cabo 6 ensayos con 3 clases distintas de proyectiles con el objetivo de poder desarrollar una aplicación en Matlab que permita ver cómo actúan los proyectiles en el interior del cuerpo humano y de esta manera poder conocer el daño que provocan.

La aplicación carga en Matlab las imágenes que han sido obtenidas de un vídeo grabado con anterioridad y cortadas con el programa Photron FASTCAM Viewer. Una vez cargadas se les ha realizado una serie de tratamientos para después recorrerlas mediante un bucle que permite calcular la posición, la velocidad, la aceleración y la energía transmitidas por el

proyectil y el área y volumen generados por este en el cuerpo. Estos datos posteriormente son grabados en un archivo Excel.

1.2 OBJETIVOS

El objetivo de este trabajo es el de diseñar una aplicación informática que sea capaz de adquirir información en tiempo real acerca del daño provocado por un proyectil en un cuerpo mediante el análisis de videos tomados.

La aplicación debe de ser capaz de analizar una secuencia de imágenes tomada a alta velocidad de cualquier tipo de proyectil para posteriormente obtener toda la información posible acerca de él y del orificio que este va dejando a su paso por el cuerpo.

1.3 ESTRUCTURA DE LA MEMORIA

La memoria del proyecto está constituida por cinco capítulos y anexos. A continuación, se describe brevemente cada parte de la memoria.

- **Capítulo 2:** Se realiza una breve introducción referente a la balística de efectos. En él se habla sobre la balística de efectos, los factores que influyen y la cavitación temporal.
- **Capítulo 3:** Se habla sobre la visión artificial, detallando su funcionamiento, sus objetivos, el procesamiento digital de imágenes y sus etapas. Los elementos que influyen y sus aplicaciones.
- **Capítulo 4:** Se realiza una introducción del programa, se detalla de una manera más profunda en que consiste el procesamiento digital de imágenes y la toolbox con la que se ha trabajado.
- **Capítulo 5:** Se definen los ensayos que se han realizado, así como los distintos proyectiles utilizados, cámaras, gelatina balística e iluminación.
- **Capítulo 6:** Se detalla el manual de funcionamiento tanto del Photron FASTCAM Viewer utilizado para cortar los videos como de la aplicación diseñada en Matlab que permite obtener características de los proyectiles.
- **Capítulo 7:** Se define la planificación que se ha llevado a cabo.
- **Capítulo 8:** Se realiza el presupuesto, detallando tanto los costes de personal como los materiales.

Aplicaciones Técnicas de Visión Artificial a Balística de Efectos

- **Capítulo 9:** Se presentan las conclusiones y las líneas futuras sobre el trabajo realizado.
- **Capítulo 10:** Se define la bibliografía que ha sido utilizada.
- **Capítulo 11:** Se adjunta en los anexos, el código del programa y los resultados para los ensayos con proyectiles SP Y HP.

2 BÁLISTICA TERMINAL O DE EFECTOS

2.1 ¿QUÉ ES?

Se trata [2] de una ciencia que se encarga del estudio del desplazamiento de los proyectiles desde el arma hasta el objetivo. La balística puede ser dividida en tres tipos:

- **Balística interna:** analiza el desplazamiento del proyectil durante el tiempo que se encuentra dentro del arma.
- **Balística externa:** analiza el desplazamiento del proyectil desde que abandona el arma hasta que alcanza el objetivo.
- **Balística terminal o de efectos:** analiza el desplazamiento del proyectil dentro de la víctima y los daños producidos.

Los dos primeros tipos pueden ser considerados exactos, mientras que la balística terminal en función de la iteración del proyectil con los fragmentos y los tejidos con los que impacta puede tener una elevada complejidad siendo muy difícil predecir el daño producido.

Cuando un proyectil penetra en un cuerpo perforándolo, pero sin orificio de salida, toda la energía cinética es transmitida a los tejidos que lo rodean. Sin embargo, cuando el proyectil atraviesa el cuerpo abandonándolo por un orificio de salida, solo parte de esta energía cinética es liberada en el cuerpo.

El daño producido por un proyectil en un cuerpo depende de varios factores, como son: la masa, la forma, la velocidad, el arrastre y la resistencia de los tejidos al paso del proyectil, las propiedades viscoelásticas de los tejidos, desplazamiento y estabilidad del proyectil dentro de los tejidos y la energía cinética liberada por el proyectil al momento del impacto. En función de la capacidad que tenga el proyectil para transmitir la energía cinética a los tejidos impactados, el daño producido por este es de mayor o de menor gravedad.

Los proyectiles pueden ser clasificados en función de la velocidad a la que salen del arma en:

- **Proyectiles de alta velocidad:** cuando viajan a una velocidad por encima de los 610 m/s.
- **Proyectiles de media velocidad:** cuando viajan a una velocidad entre los 335 m/s y los 610 m/s.

- **Proyectiles de baja velocidad:** cuando viajan a una velocidad por debajo de los 335 m/s.

En función de la velocidad que tengan los proyectiles, producirán una destrucción mayor o menor, lo que supone un tratamiento distinto. Además, los proyectiles de alta velocidad producen una mayor destrucción debido a la fragmentación y los efectos de la cavitación temporal (*Figura 2-1*).

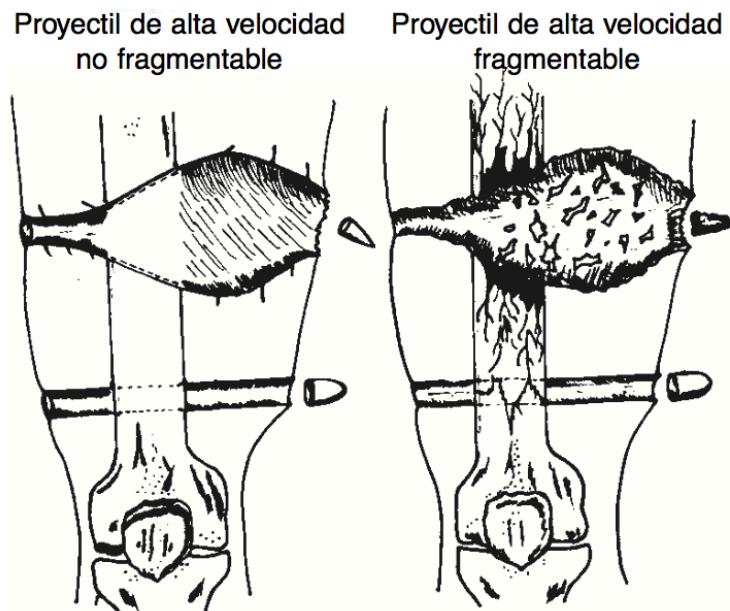


Figura 2-1: Lesiones producidas por los proyectiles de alta velocidad (arriba) y los de baja velocidad (abajo).

2.2 FACTORES QUE INFLUYEN EN LA BALÍSTICA TERMINAL

Existen una serie de factores que influyen en el comportamiento del proyectil y la iteración de este sobre el blanco, estos factores son los siguientes:

2.2.1 Velocidad del proyectil

Velocidad de impacto

Se trata de la velocidad [5] a la que se desplaza el proyectil en el momento del choque. Aunque, es considerado el factor más importante en la capacidad que tiene el proyectil para herir, también hay que tener en cuenta otros aspectos como son las características del tejido.

Velocidad residual o remanente

Se trata de la velocidad [5] que tiene el proyectil una vez que ha perforado el blanco.

La velocidad residual depende de la dureza del objetivo, del ángulo de impacto, del tipo de tejido, etc. Debido a estos factores puede darse la situación de que dos proyectiles disparados sobre un mismo cuerpo tengan distintos efectos, como pueden ser que uno atraviese el cuerpo y el otro no.

2.2.2 Ángulo de incidencia

Se llama así al ángulo [3] con el cual el proyectil penetra en el blanco. Este ángulo, representado por la letra griega θ , se denomina al formado entre la línea perpendicular a la superficie del blanco en el punto de impacto y la línea de impacto del proyectil (*Figura 2-2*).

Este ángulo de impacto depende de una serie de variables como pueden ser la estabilidad del proyectil durante el vuelo, características fisicoquímicas del proyectil y del blanco, que condicionan la existencia o no de penetración en el blanco.

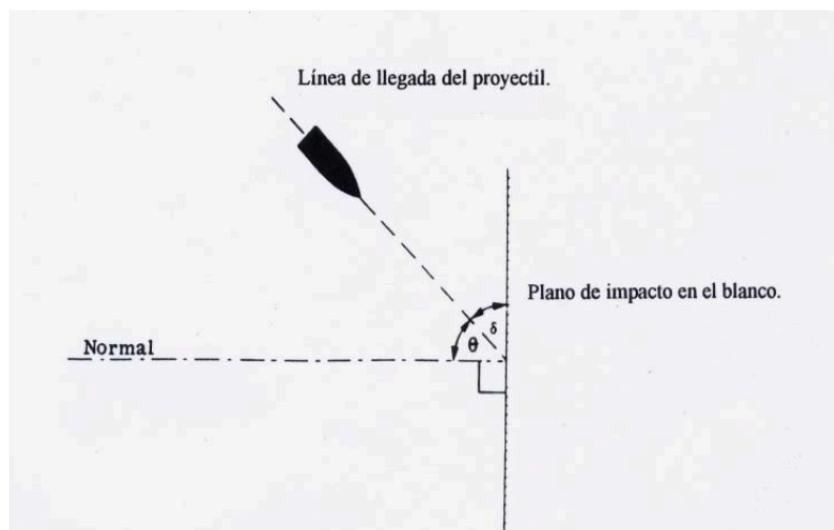


Figura 2-2: Ángulo de incidencia de un proyectil en un cuerpo.

2.2.3 Características del proyectil

La forma [3] y el diseño de la munición es otro factor importante que afecta a la iteración del proyectil sobre el blanco.

En función del diseño de la ojiva del proyectil, este genera distintos tipos de efectos. Las ojivas pueden ser:

- **Afilada o en punta:** son balas cuya razón entre la longitud de la ojiva y el calibre ≥ 1 . La acción llevada a cabo por este tipo de proyectil es, según su eje axial, de taladro.

- **Roma:** son balas cuya razón entre la longitud de la ojiva y el calibre < 1. La acción que lleva a cabo es la de arrancar material perteneciente al blanco, transmitiendo de esta manera su cantidad de movimiento. Este fenómeno es conocido como plugging (*Figura 2-3*).

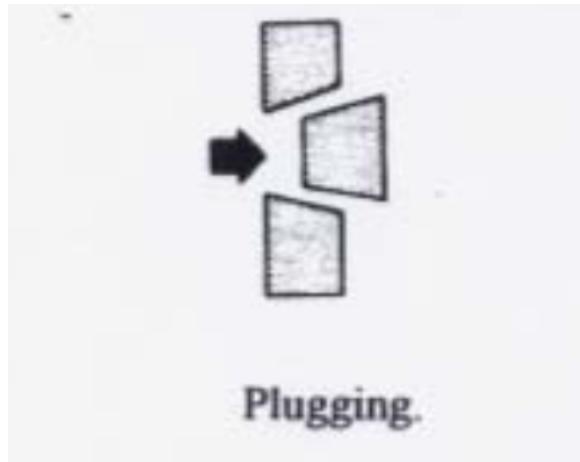


Figura 2-3: Efecto plugging, debido a balas tipo romo.

Aunque, a priori puede parecer que un proyectil afilado o en punta, junto con un material de dureza y densidad elevada, puede ser el que presente unas mejores características a la hora de maximizar el efecto penetrador del proyectil, esto puede dar lugar a una serie de problemas que impidan un aumento en la longitud del proyectil llevado a cabo de forma arbitraria. Estos problemas son:

- En el momento que se lleve a cabo el impacto, se produzca un fallo a flexión y pandeo del proyectil, dando lugar a una posible rotura antes de haber sido capaz de transmitir su cantidad de movimiento sobre el objetivo, produciendo únicamente daños superficiales.
- Durante el vuelo hasta llegar a alcanzar el objetivo, se produzca una desestabilización, dando lugar a una falta de precisión necesaria para alcanzar al blanco.

2.2.4 Distancia de disparo

En función [4] de la distancia desde la cual se realice el disparo, puede ser clasificado en:

Disparo a bocajarro

Se trata de un disparo en el cual el cañón del arma de fuego es apoyado contra el cuerpo. Este tipo de disparo deja una serie de rasgos característicos en el cuerpo receptor del disparo, que son los siguientes:

- Suele dejar la marca de la trompetilla del arma.
- Puede ocurrir, que cuando el disparo se produce sobre una parte dura (por ejemplo, un hueso), las dimensiones del orificio de entrada sean mayores que las del orificio de salida, debiéndose esto a la explosión que se produce en la trompetilla al abandonar la bala el arma.
- Cuando el disparo es realizado sobre una parte blanda, puede parecer que se trata de un disparo de larga distancia. Sin embargo, al examinarlo, se observa que hay residuos de pólvora en la piel y en los músculos, lo que da a entender que el disparo ha sido a bocajarro.

Disparo a quemarropa

Debe su nombre al hecho de que por regla general quema la ropa. Se realiza a una distancia inferior a los diez centímetros y es el disparo que permite observar el tatuaje de la manera más sencilla.

Disparo a corta distancia

Considerada la máxima distancia a la cual el disparo deja tatuaje. Se trata de una herida, por lo general limpia, que se realiza a una distancia entre los diez centímetros y un metro.

Disparo a larga distancia

Realizado a una distancia superior de un metro. Todo disparo que no deja tatuaje en el objetivo es considerado disparo de larga distancia.

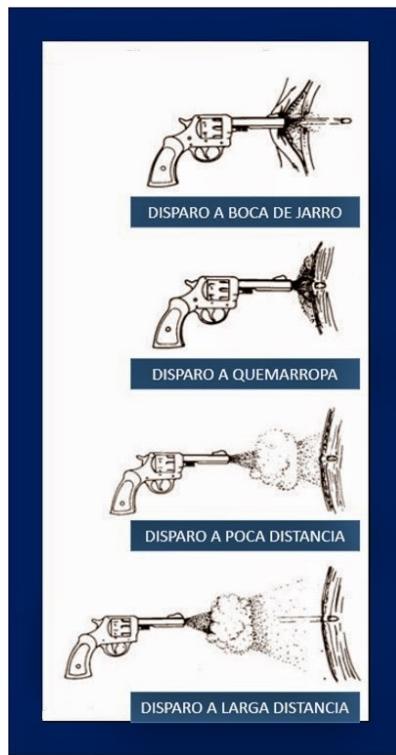


Figura 2-4: Distancia de disparo.

2.3 CAVITACIÓN TEMPORAL DEL PROYECTIL

Debido [3] a la penetración del proyectil a alta velocidad en un cuerpo, el momento, la velocidad y el giro de este es transmitido a las capas más cercanas del blanco. Estas capas, como consecuencia del momento transferido, comienzan a moverse y a oscilar en función del avance del proyectil, dando lugar a una zona de inestabilidad (*Figura 2-5*), que por espacio de unos milisegundos adquiere un movimiento armónico con una pulsación determinada aunque variable en función del avance del proyectil, dando lugar a expansiones y compresiones hasta volver a su estado de inestabilidad.

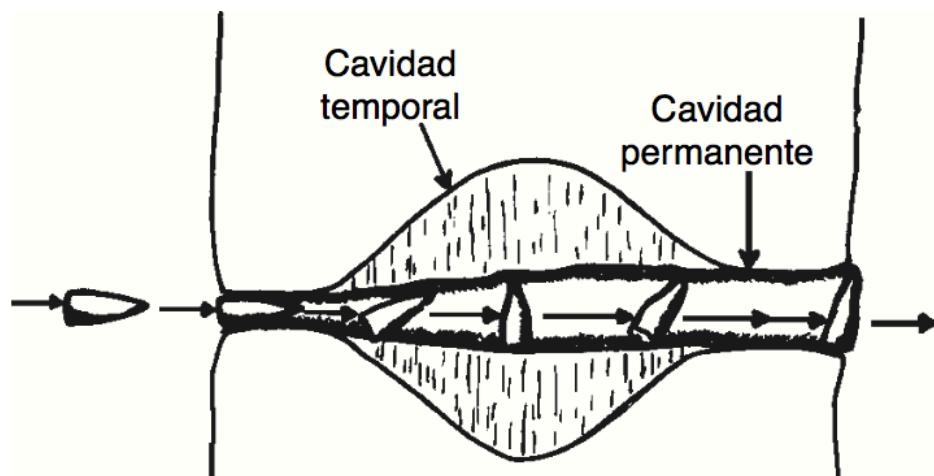


Figura 2-5: Cavitación temporal producida por un proyectil [2].

Como consecuencia del orificio originado por el proyectil, pueden aparecer fenómenos de corrosión y esto es debido a que en el orificio originado existe una presión inferior a la atmosférica, lo que provoca un fenómeno de succión dando lugar a que partículas extrañas al blanco penetren en la cavidad.

La cavitación es producida mayoritariamente tras el paso del proyectil, con una gran dependencia de la velocidad de este. Esto es debido a que contra mayor velocidad, mayor energía cinética, dando lugar a una mayor cantidad de energía transmitida a las capas que lo rodean y a una mayor cavitación temporal y por lo tanto un mayor daño producido.

Desde la *Figura 2-6* hasta la *Figura 2-10*, se puede observar el orificio producido en la gelatina balística por un proyectil a alta velocidad. En estas imágenes podemos ver desde que se produce el impacto, pasando por la penetración, la cavitación y la salida del proyectil, con los distintos estados de vibración en el material.

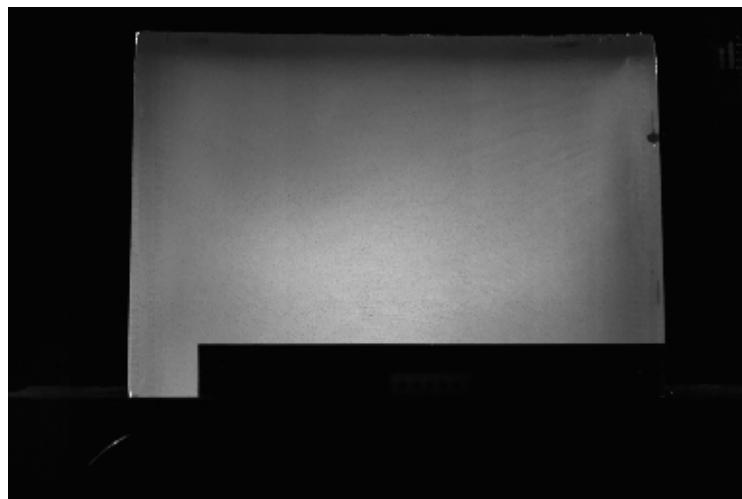


Figura 2-6: Proyectil fmj impactando con la gelatina balística.

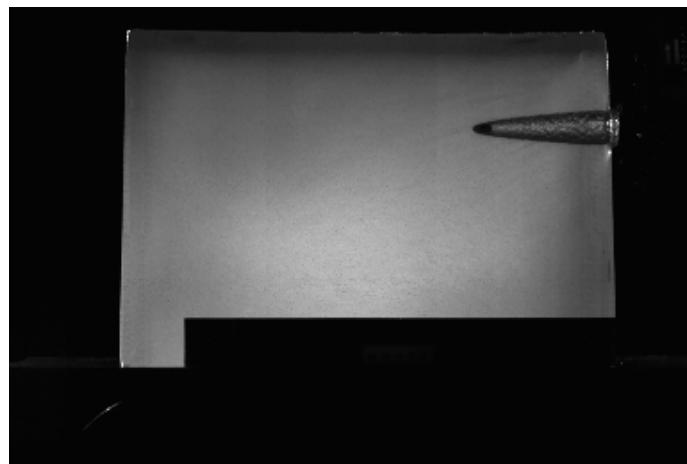


Figura 2-7: Proyectil fmj penetrando en la gelatina balística.

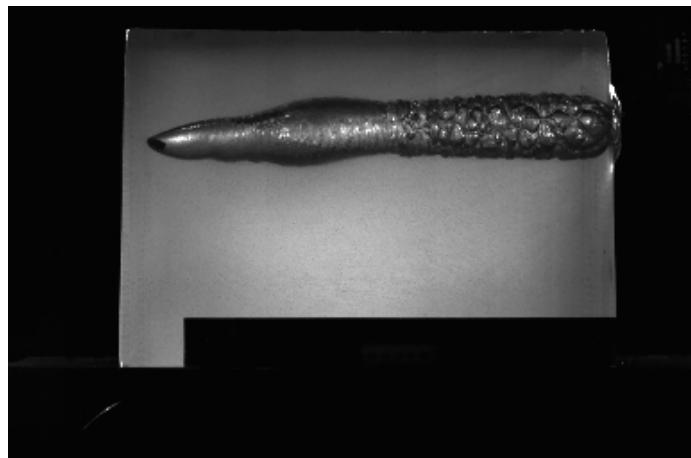


Figura 2-8: Proyectil fmj avanza por el interior de la gelatina balística.

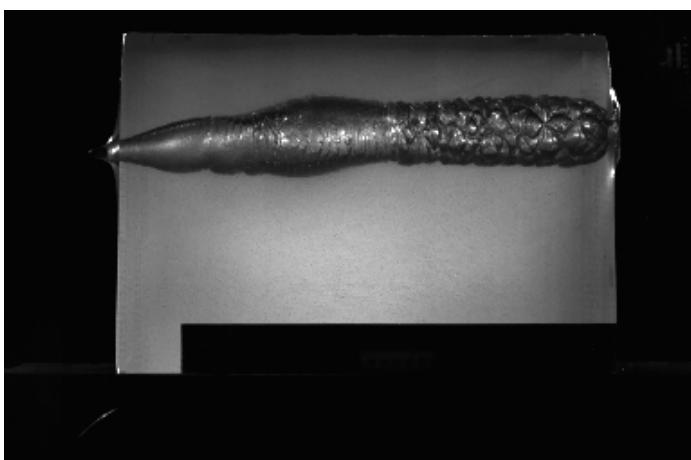


Figura 2-9: Salida proyectil fmj de la gelatina balística.



Figura 2-10: Crecimiento del orificio generado por el proyectil debido a la cavitación temporal.

3 VISIÓN ARTIFICIAL

3.1 ¿CÓMO FUNCIONA?

Mediante el uso de cámaras [1][6] basadas en matrices de sensores sensibles a la luz (CMOS o CCD), se captan imágenes en línea para posteriormente, con el uso de técnicas de análisis de imagen, ser tratadas.

La visión artificial tiene una gran cantidad de ventajas, entre las que cabe destacar:

- Permite realizar tareas de inspección repetitivas realizadas por operarios.
- Permite llevar a cabo controles de calidad que no eran posibles llevar a cabo por medio de métodos tradicionales.
- Permite inspeccionar objetos sin ser necesario el contacto físico.
- Permite inspeccionar a gran velocidad el 100% de la producción.
- Permite reducir el tiempo de los ciclos en procesos que estén automatizados.
- Permite realizar inspecciones en procesos donde existe una gran diversidad de piezas con cambio frecuentes de producción.

Se debe de tener en cuenta tres aspectos clave en el ámbito de la visión artificial:

- Se trata de un proceso computacional.
- El observador debe ser quien defina qué información obtener.
- Eliminar la información que no resulte útil.



Figura 3-1: Visión artificial frente a la visión humana.

3.2 OBJETIVOS

Entre los objetivos que debe de ser capaz de cumplir la visión artificial se encuentran los siguientes:

- Ser capaz de detectar, segmentar, localizar y reconocer objetos en imágenes.
- Ser capaz de evaluar los resultados obtenidos.
- Ser capaz de registrar diferentes imágenes de una misma escena u objeto.
- Ser capaz de seguir un mismo objeto durante una secuencia de imágenes.
- Ser capaz de mapear una escena con el objetivo de generar un modelo tridimensional de la misma.

Mediante el reconocimiento de patrones, aprendizaje estadístico, geometría de proyección, procesamiento de imágenes, teoría de grafos y otros campos, la visión artificial es capaz de lograr estos objetivos.

3.3 PROCESAMIENTO DIGITAL DE LAS IMÁGENES Y ETAPAS QUE LO COMPONENTE

La digitalización de una imagen consiste en transformarla del mundo real a un formato que permita al ordenador realizar un procesamiento de la imagen.

El procesamiento digital de cualquier imagen está constituido por un conjunto de etapas que se pueden ver en la *Figura 3-2* y que van a ser explicadas a continuación:

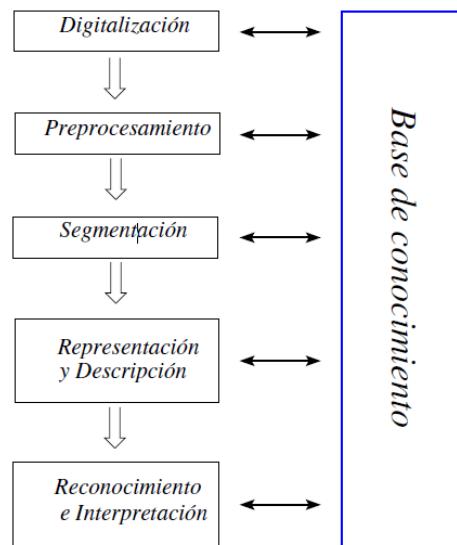


Figura 3-2: Etapas del procesamiento digital de las imágenes.

1. **Digitalización:** Se trata de la primera etapa en el procesamiento de imágenes y en ella se lleva a cabo el proceso de transformar la imagen analógica a digital.

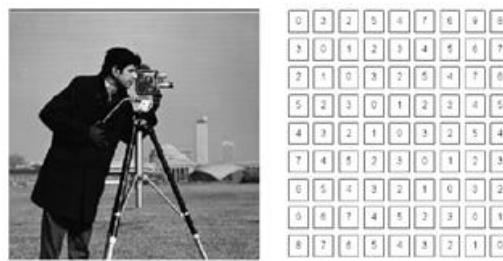


Figura 3-3: Imagen analógica frente a imagen digitalizada.

2. **Pre-procesamiento:** Consiste en un acondicionamiento de las imágenes que permite una mayor probabilidad de éxito en las etapas sucesivas.
3. **Segmentación:** Consiste en un proceso de obtención de la información contenida en la imagen.
4. **Representación y descripción:** Consiste en parametrizar los objetos o partes que han sido generadas en la etapa de segmentación mediante el uso de sistemas de coordenadas cartesianas o polares.
5. **Reconocimiento:** En esta etapa, mediante el uso de los descriptores de los distintos objetos, se lleva a cabo una clasificación.

6. Interpretación: Consiste en interpretar los resultados obtenidos de los objetos reconocidos.

3.4 ELEMENTOS

En un sistema de visión artificial intervienen una serie de elementos que afectan al funcionamiento de este y son los siguientes:

3.4.1 Iluminación del sistema de visión artificial

La iluminación, dentro de las aplicaciones de la visión artificial, tiene gran importancia, aunque en infinidad de ocasiones es subestimada en detrimento de otros elementos como pueden ser las lentes, las cámaras o los framegrabbers.

En realidad, debe de ser considerada como la parte más crítica, debido a que debe estar optimizada al máximo para que de esta manera las cámaras, las cuales no son tan sensibles ni versátiles con el ojo humano, sean capaces de capturar una imagen que en el caso del ojo humano podría ser capturada sin la necesidad de una iluminación tan especializada.

Las cámaras, lo que hacen, es captar la luz reflejada por los objetos, por lo que es muy importante ajustarla en función del objeto que vayamos a iluminar, ya que una bola de acero refleja la luz de manera distinta a una hoja de papel en blanco.

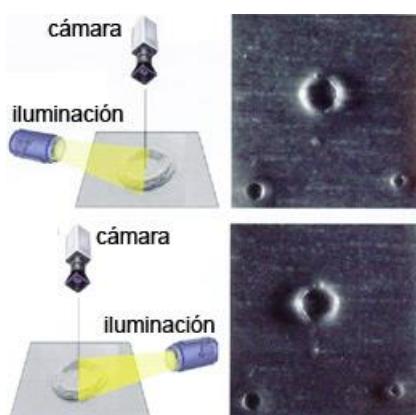


Figura 3-4: Importancia de iluminación a la hora de resaltar defectos superficiales.

En cuanto a las ventajas del uso de una iluminación adecuada frente a una iluminación incorrecta, son las siguientes:

- El sistema es capaz de resolver el problema de una manera más sencilla.

- Se utiliza un menor número de filtros de software lo que permite llevar a cabo el proceso de una manera más rápida.

3.4.2 Ópticas o lentes

Las ópticas o lentes tienen como objetivo obtener una imagen enfocada de uno o varios objetos por medio de la transmisión de luz al sensor de la cámara de forma controlada.

3.4.3 Cámara: definición y características.

La cámara tiene como función el transferir al sistema electrónico la escena capturada en el sensor de imagen con la ayuda de las lentes.

El tipo de cámaras, que son utilizadas en la visión artificial, son más sofisticadas que las cámaras convencionales, ya que deben de ser capaces de realizar un control completo de la sensibilidad, de la velocidad de obturación, de los tiempos y señales y de otros factores que son fundamentales a la hora de ser integrados en un sistema de visión artificial.

3.4.4 Programación

El objetivo de la programación es el de dar un conjunto de instrucciones que permita realizar la secuencia de trabajo deseada. Para poder llevar a cabo esta programación es necesario conocer de forma detallada el proceso que debe ser controlado y una unidad de control tanto para el sistema de iluminación, como para la cámara y su correspondiente software de programación.

La unidad de programación consiste por tanto en una interfaz física entre el usuario y el sistema de visión, que permite la transferencia de programas, la supervisión y el control del proceso. El lenguaje de programación consiste en un conjunto de símbolos y textos que permiten al usuario codificar las leyes de control deseadas y que sean entendibles por la unidad de programación. Puede estar basado en:

- Texto: constituido por un texto estructurado o por una lista de instrucciones.
- Gráfico: constituido por diagrama de bloques funcionales o diagrama de escalera.

En la *Figura 3-5* se puede ver un diagrama de bloques funcionales constituido por la siguiente secuencia:

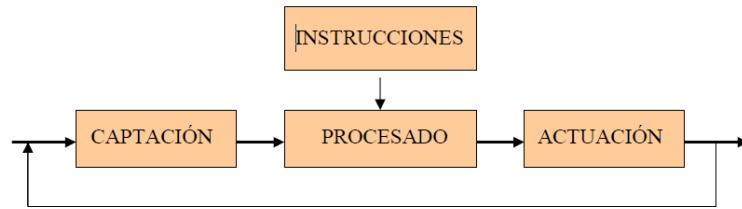


Figura 3-5: Proceso operativo del sistema.

- **Captación:** En este bloque se obtiene la imagen del objeto que va a ser inspeccionada.
- **Instrucciones:** En este bloque se definen el conjunto de operaciones que son necesarias para resolver el problema.
- **Procesado:** En este bloque se lleva a cabo, a partir de las instrucciones aplicadas, un tratamiento de la imagen.
- **Actuación:** En este bloque, en base al resultado obtenido, se actúa sobre el entorno (aparato, pieza, elemento).

3.4.5 Hardware: Framegrabbers

La mayoría de las cámaras utilizadas actualmente en los sistemas de visión artificial y análisis de la imagen son conectadas de forma directa al ordenador por medio de diferentes puertos: FireWire, USB, Gigabit Ethernet, etc. No obstante, algunos tipos de cámaras aun necesitan conectarse al ordenador mediante una tarjeta de procesado de imágenes (IMAQ) o una placa de captura (framegrabbers o tarjetas de adquisición de imágenes). Son aquellas que tienen interfaces de conexión cameralink o coaxpress.

Los framegrabbers se encargan de la captura en las aplicaciones de visión industrial y de análisis de imagen, donde se necesita capturar imágenes con la mayor calidad posible, enviándolas a la memoria del ordenador donde son procesadas, analizadas y visualizadas.

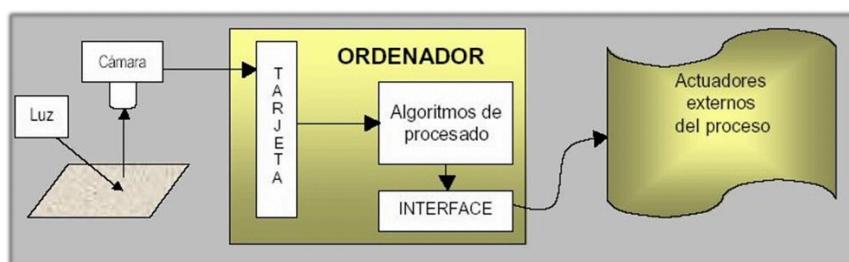


Figura 3-6:Etapas del proceso de captura de imágenes.

A la hora de escoger el framegrabber es necesario conocer en que va a ser utilizado debido a que los precios y especificaciones de las tarjetas varían enormemente.

3.4.6 Softwares empleados en la visión artificial

Actualmente, con la evolución de los procesadores de los PCs y la llegada de los nuevos tipos de buses se es capaz de visualizar las imágenes en tiempo real y llevar a cabo los procesos en tiempos suficientemente cortos, lo que permite resolver las aplicaciones de visión artificial en tiempo real tanto en entornos industriales como científicos.

El objetivo del software es el de interpretar y analizar los pixeles. El resultado final puede ser desde la medida de una partícula, pasando por la determinación o lectura de una serie de caracteres, o cualquier otro proceso que podamos imaginar. En función de si la aplicación se lleva a cabo en un entorno industrial o científico los pasos a seguir para su resolución son distintos:

- **APLICACIONES INDUSTRIALES:**
 - Captura de la imagen.
 - Definición de la región de interés donde se realizarán las medidas.
 - Inicialización de las tolerancias para determinar si la pieza es o no correcta.
 - Ejecutar las medidas.
 - Generar una salida apropiada.
- **APLICACIONES CIENTÍFICAS:**
 - Captura de la imagen.
 - Realizar proceso de mejora.
 - Determinación de los elementos a medir.
 - Almacenar las medidas y realizar procesos gráficos o estadísticos.

3.5 APLICACIONES

En la *Figura 3-7*, se pueden ver las distintas aplicaciones en las cuales es utilizada la visión artificial:

Aplicaciones Técnicas de Visión Artificial a Balística de Efectos

Militares	Agricultura
Detección y seguimiento de objetivos	Análisis de las plantaciones: crecimiento, enfermedades.
Análisis del terreno	Análisis de imágenes tomadas por satélites
Armas inteligentes.	
Control de calidad	
Verificación de etiquetas	Análisis de imágenes tomadas por rayos x
Inspección de containers	Análisis de imágenes tomadas por ultrasonidos
Inspección de motores	
Inspección de cristales	
Control de calidad de comida	Análisis de sangre
Inspección de soldaduras	Análisis de DNA
Inspección de circuitos impresos	
Inspección de madera, tela, fundiciones, papel	Identificación
	Identificación automática de huellas dactilares.
	Reconocimiento de caras
Robótica	
Guiado de robots industriales	Seguridad
Navegación de robots móviles	Vigilancia de edificios
	Detección de explosivos por rayos x
	Control de tráfico
	Identificación de matrículas de vehículos
	Control del tráfico viario

Figura 3-7: Aplicaciones visión artificial [1].

4 MATLAB Y PROCESADO DIGITAL DE IMÁGENES

4.1 ¿QUÉ ES MATLAB?

MATLAB (Matrix LABoratory) [8] [9] [14], disponible para Windows, Mac OS X y Unix, es un software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio.

Las funciones básicas de Matlab permiten entre otras cosas: manipular matrices, representar datos y funciones, implementar algoritmos, crear interfaces de usuario (GUI) y comunicarse con programas en otros lenguajes y otros dispositivos hardware.

Además, Matlab tiene una serie de herramientas que aumentan su utilidad, como son:

- **Simulink:** es una plataforma de simulación multidomínio.
- **Guide:** editor de interfaces de usuario (GUI).
- **Toolbox:** son cajas de herramientas que cubren las áreas principales del mundo de la ingeniería y la simulación. Ejemplo: *toolbox* de procesado digital de imágenes, *toolbox* de procesado de señal, *toolbox* de comunicaciones... y así hasta más de 150 herramientas.

Matlab utiliza un lenguaje de programación similar a los lenguajes tradicionales: Fortran, Visual Basic o C. Sin embargo, el elemento básico de trabajo son las matrices, lo que permite a Matlab una mayor velocidad de resolución numérica de los problemas. El trabajo con matrices aporta además una gran flexibilidad y una gran eficiencia matemática a la hora de describir variables. Por ejemplo, una imagen puede ser escrita como una matriz de pixeles, un sonido como una matriz de fluctuaciones, y por regla general cualquier relación lineal entre los componentes de un modelo matemático puede ser descrito con una matriz.

Aunque Matlab comenzó a utilizarse por investigadores y profesionales pertenecientes al sector de la ingeniería de control, se consiguió expandir rápidamente a otros sectores. En la actualidad, tiene una gran acogida dentro de la enseñanza del álgebra lineal y el análisis numérico, además de ser muy utilizado por científicos que se dedican al procesado digital de imágenes. Esta rápida expansión se debe a la efectividad de la política de licencias de evaluación a un bajo coste.

4.2 INTRODUCCIÓN AL PROCESADO DIGITAL DE IMAGENES

4.2.1 Definición

El procesado digital [13] de imágenes consiste en el almacenamiento, transmisión y representación de la información de las imágenes digitales con la ayuda de una computadora.

El término imagen hace referencia a una función bidimensional de intensidad de luz $f(x, y)$ donde “x” e “y” hacen referencia a las coordenadas espaciales y el valor de f en cualquier punto (x, y) es proporcional a la intensidad de la imagen en ese mismo punto. Una imagen digital puede ser escrita como una matriz cuyos índices de fila y columna hacen referencia a un punto de la imagen cuyo valor coincide con el nivel de intensidad de la luz en ese punto. Cada elemento perteneciente al array coincide con un elemento de la imagen y es denominado pixel.

El procesado digital de imágenes tiene como principales intereses los siguientes aspectos:

- Una mejora de la información que contiene una imagen que permita un fácil entendimiento.
- Favorecer la percepción autónoma por parte de una máquina por medio del tratamiento de datos de una escena.

A continuación, se realiza una explicación que permite entender mejor las diferencias entre los siguientes términos que suelen ser confundidos con frecuencia:

- **El análisis de imágenes:** consiste en la extracción de información cuantitativa de la imagen, dando como resultado una tabla de datos, una gráfica o cualquier tipo de representación de los datos numéricos.
- **El procesado de imágenes:** consiste en un proceso de mejora de la calidad de la imagen que permite apreciar mejor determinados detalles.
- **La visión por computador o visión artificial:** consiste en una rama de la inteligencia artificial cuyo objetivo es emular a la visión humana por medio de la programación de un computador para que sea capaz de entender una escena o las características de una imagen.

En base a los tipos de procesos implicados en estas disciplinas, se suele realizar la siguiente clasificación:

- **Los procesos de nivel bajo (procesado):** se lleva a cabo una reducción del ruido, la mejora del contraste, en general, una mejora de las características de la imagen, en donde todas las entradas/salidas son imágenes.
- **Los procesos de nivel medio (análisis):** consiste en analizar los niveles bajos e incluir la segmentación, descripción de objetos, clasificación, etc. La entrada es una imagen mientras que la salida son atributos de los objetos (bordes, contornos, identidades de objetos individuales).
- **Los procesos de nivel alto (interpretación):** orientados hacia la interpretación de los elementos obtenidos en los anteriores niveles, siendo necesario el entendimiento y la toma de decisiones en función del contenido observado.

En base a esto, se define procesado digital de imágenes a los procesos en los cuales las entradas y salidas son imágenes (bajo nivel) y aquellos procesos que consiguen extraer atributos de las imágenes (nivel medio).

4.2.2 Historia

Aunque se ha llevado a cabo una continua miniaturización de componentes y desarrollo de sistemas de almacenamiento, a continuación se detallan una serie de momentos importantes en el procesado digital de imágenes:

- 1921: inicio del procesado digital de imágenes por medio de la impresión de periódicos. Los datos eran codificados y transmitidos por medio de un cable submarino entre Londres y Nueva York, donde se llevaba a cabo la reconstrucción e impresión de la información.
- 1922: se produce una mejora del proceso con la implantación de una técnica cuya base era la reproducción fotográfica por medio de cintas perforadas en las terminales telegráficas receptoras lo que permitía obtener 5 escalas de gris.
- 1929: una nueva mejora en la técnica permitía reproducir las fotográficas hasta obtener 15 niveles de gris. Esto todavía no era considerado el inicio del procesado digital de imágenes debido a que no se usaba la computadora.
- 1948: invención del transistor en los Laboratorios Bell.

- 1958: invención del circuito integrado por Texas Instrument.
- Años 60: se comienzan a ver las primeras técnicas de procesado digital de imágenes por computador gracias a la llegada de los grandes computadores junto al programa espacial americano.
- 1964: las imágenes de la Luna enviadas por el satélite Ranger 7 son procesadas para corregir distorsiones propias de la cámara en el Laboratorio de Propulsión de la NASA.
- Años 70: desarrollo de los sistemas operativos.
- Años 80: desarrollo de lenguajes de programación de alto nivel.
- 1981: introducción del ordenador personal (IBM).

4.2.3 Áreas de aplicación

A pesar de que al principio las técnicas de procesado digital de imágenes solo eran aplicadas en programas espaciales, con el paso del tiempo empezaron a aplicarse en gran cantidad de campos y problemas en los cuales era necesario llevar a cabo una mejora en las imágenes que facilitase la interpretación y el análisis.

En la actualidad el procesado digital de imágenes es utilizado en una gran cantidad de áreas de forma habitual, para saber cuáles son las áreas implicadas existen diferentes tipos de fuentes (acústica, electrónica), siendo la principal fuente de energía el espectro electromagnético (*Figura 4-1*).

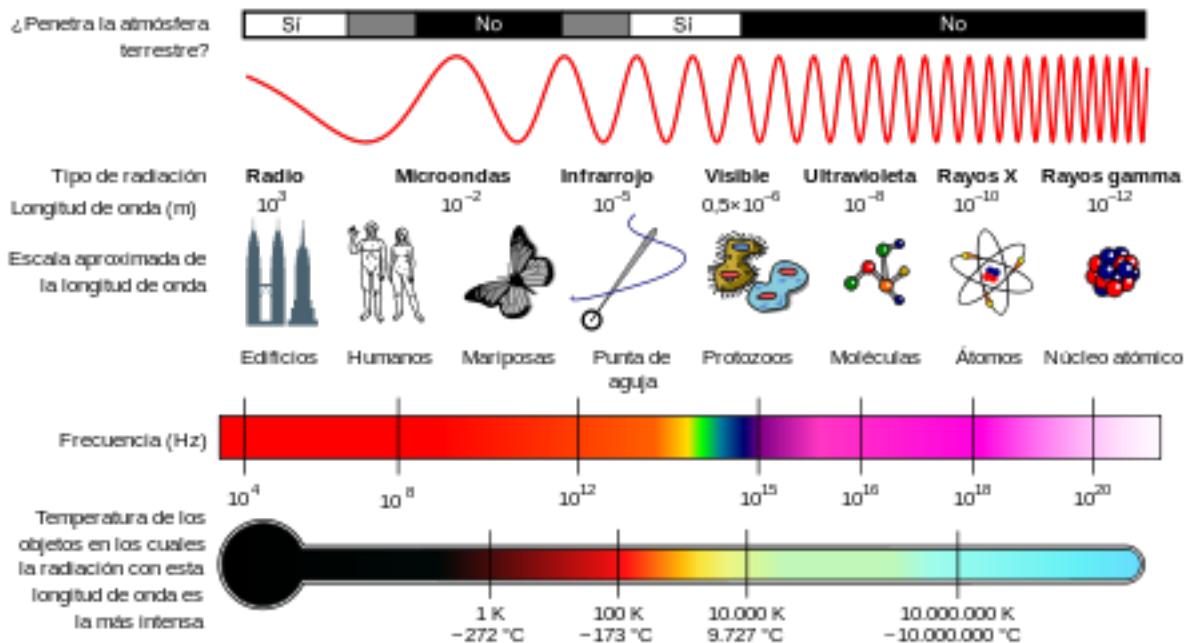


Figura 4-1: Espectro electromagnético [7].

Rayos Gamma

Los rayos gamma son utilizados para la captura de imágenes en las siguientes aplicaciones:

- Tomografía por Emisión de Positrones (PET) (Figura 4-2): consiste en inyectar al paciente un isótopo radiactivo, el cual emite rayos gamma que son capturados posteriormente.
- Astronomía: permite detectar la emisión de rayos gamma procedentes de los astros.

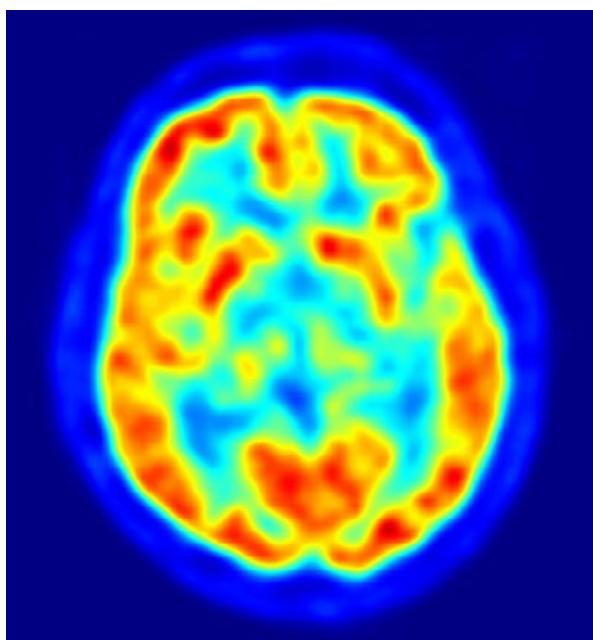


Figura 4-2: Escáner PET del cerebro humano.

Rayos X

Las aplicaciones en las cuales son utilizados los rayos X para la captura de imágenes son las siguientes:

- La medicina: radiografía, angiografía, Tomografía Axial Computarizada (TAC).
- La astronomía.
- La industria.
- En el control de calidad de placas de circuito impreso o de alimentos.

Banda Ultravioleta

La banda ultravioleta es utilizada para la obtención de imágenes, siendo las principales aplicaciones las siguientes:

- En litografía.
- En inspección industrial.
- En microscopía.
- En láseres.
- En imágenes biológicas.
- En astronomía.

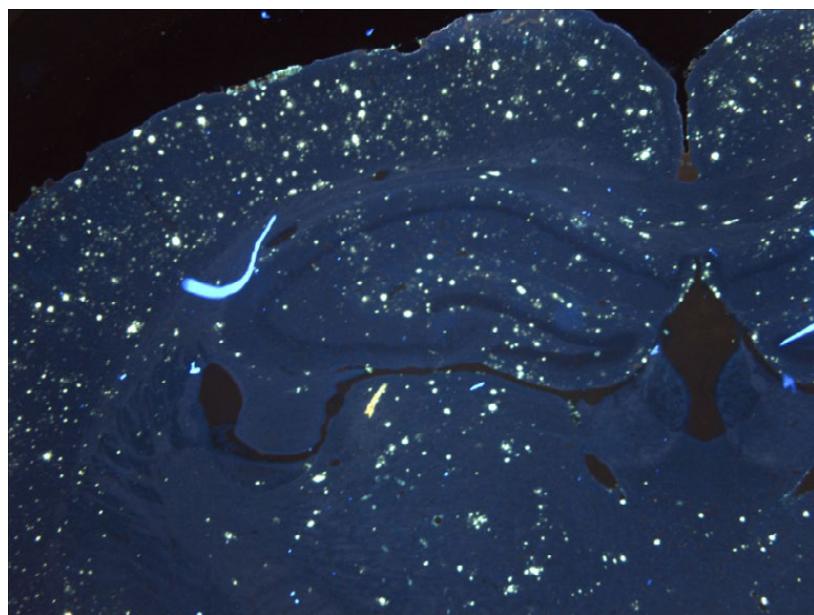


Figura 4-3: Cerebro de un ratón con luz ultravioleta..

Banda visible e infrarroja

El uso conjunto de la banda visible e infrarroja aporta el mayor número de aplicaciones, siendo estas:

- En microscopía.
- En astronomía.
- En detección remota.
- En controles de calidad en la industria y en las aplicaciones de identificación de la policía.



Figura 4-4: Detección de vegetación con imagen infrarroja.

Banda de microondas

La principal aplicación de las imágenes de la banda de microondas es el radar, ya que este puede captar datos de cualquier región y en cualquier momento, independientemente de las condiciones de clima o de luz ambiental, mediante el uso de los pulsos del microondas que permiten capturar una imagen a partir de la energía reflejada.

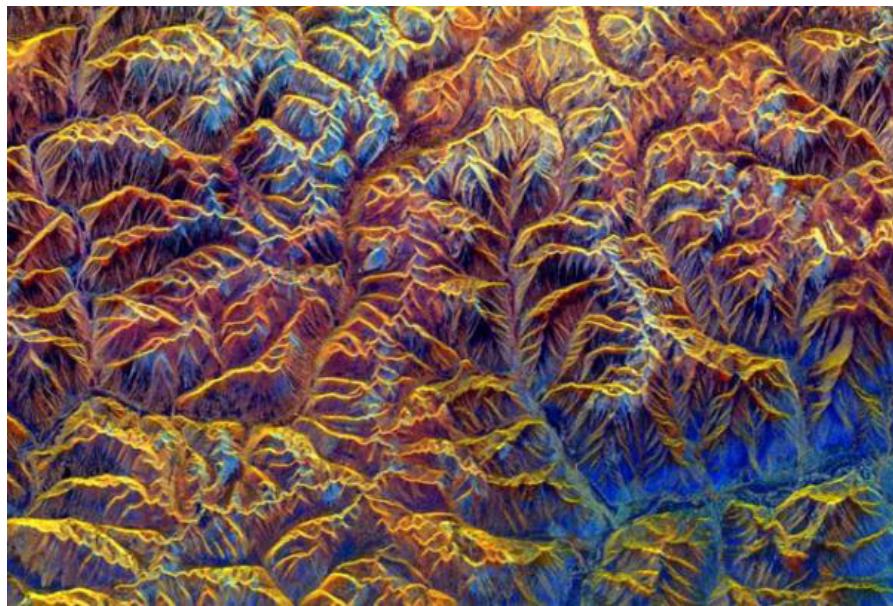


Figura 4-5: Montañas en el Tibet vistas por un radar topográfico.

Banda de radio

Las imágenes captadas con el uso de la banda de radio se utilizan en las siguientes aplicaciones:

- Astronomía.
- Medicina: Imágenes de resonancia magnética (MRI).

Ultrasonidos

Con el uso del sonido también se pueden obtener imágenes, siendo estas imágenes acústicas.

Principales aplicaciones:

- Medicina: seguimiento del embarazo de un bebe o la captación de imágenes del cuerpo, como el tiroides o las capas musculares.
- Explotaciones geológicas: petróleo y minerales.
- En la industria.



Figura 4-6: Detección de emisiones de gases en el fondo del Golfo de México con sonar.

Rayos de electrones

Mediante el uso de rayos de electrones utilizados para la obtención de imágenes electrónicas, como las de la microscopía electrónica, la microscopía de escaneo de electrones (SEM) y la microscopía de transmisión de electrones (TEM), existe la posibilidad de captar un filamento de tungsteno o detectar un circuito integrado dañado.

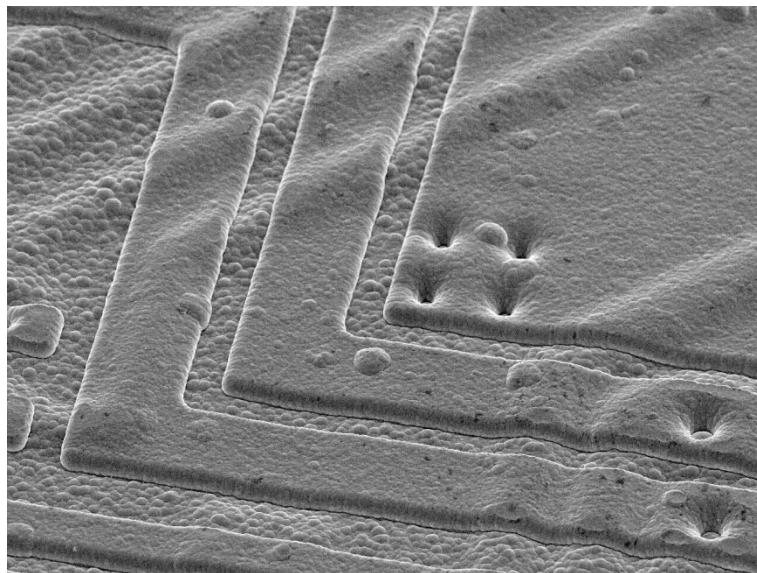


Figura 4-7: Circuito integrado ampliado 2000 veces con SEM.

4.2.4 Clases principales de procesado digital de imágenes

Es conveniente llevar a cabo una clasificación de los distintos procesos que participan en el procesado digital de imágenes y de esta manera conseguir tener una visión general más estructurada. Estos procesos pueden ser agrupados en tres grandes clases:

- **Mejora o realce de la imagen:** consiste en realizar un procesado de la imagen que aporte un resultado más apropiado para una determinada aplicación, siendo las tareas más habituales en esta clase:
 - La mejora de la nitidez o el aclarado en imágenes desenfocadas.
 - La eliminación del ruido.
 - La mejora del contraste y del brillo.
 - La detección de bordes.
- **Restauración de la imagen:** consiste en la eliminación del daño producido en una imagen debido a una causa determinada, siendo las tareas más habituales en esta clase:
 - Eliminación del desenfoque por movimiento.
 - Eliminación de interferencias periódicas.
 - Eliminación de distorsiones ópticas.
- **Segmentación de la imagen:** se encarga de dividir la imagen en trozos o de aislar objetos dentro de una imagen. Este tipo de procesado, dependiendo del objetivo y del problema de cada caso puede aparecer o no. Las tareas más habituales en esta clase son:
 - Búsqueda y selección de formas determinadas en la imagen.
 - Mascaras de la imagen.

4.2.5 Pasos fundamentales en el procesado digital de imágenes

No todas las imágenes deben ser procesadas de la misma manera para resolver el problema, esto depende de su complejidad. Sin embargo, por regla general suelen seguir los siguientes pasos:

1. **Adquisición de la imagen:** consiste en la digitalización de las imágenes con la ayuda de sensores. Generalmente esta etapa suele ir acompañada de un pre-procesado de la imagen, como puede ser un escalamiento de la imagen.
2. **Mejora de la imagen:** consiste en llevar a cabo una serie de procesos de manipulación de las imágenes, para de esta manera lograr un resultado más adecuado que el obtenido en el original para una determinada aplicación. Por ejemplo: obtención de detalles que en la imagen original no se veían.

3. **Restauración de la imagen:** al igual que el paso anterior permite realizar una mejora de la imagen. Sin embargo, a diferencia de la mejora que se trata de una etapa subjetiva, la restauración es objetiva debido a que las técnicas utilizadas en esta etapa suelen ser modelos matemáticos o probabilísticos de degradación de la imagen.
4. **Procesado del color:** en función de la imagen, esta puede ser capaz de dar mayor información con un color que con otro.
5. **Ondículas (wavelets):** utilizadas sobre todo en compresión, permiten una representación de las imágenes en distintos grados de resolución.
6. **Compresión:** permite una reducción del almacenamiento requerido para guardar una imagen, o del ancho de banda para tu transmisión.
7. **Operaciones morfológicas:** se trata de herramientas que tienen una gran utilidad a la hora de extraer componentes útiles de la imagen para una posterior representación y descripción de formas.
8. **Segmentación:** se trata de una de las tareas más complicada en el procesado digital de imágenes. En ella la imagen es dividida en sus partes constituyentes, extrayéndose o aislando los objetos para más tarde ser analizados.
9. **Representación y descripción:** en esta etapa, la imagen es recibida de forma segmentada constituida únicamente por regiones o fronteras. Además, se decide si debe ser tratada como una frontera o una región, extrayéndose los atributos que pueden constituir información de interés o que son básicos para diferentes clases de objetos.
10. **Reconocimiento:** es la etapa en la cual se le asigna una etiqueta a un objeto en función de su descripción o se le da un significado a un conjunto de objetos los cuales ya han sido reconocidos previamente.

4.3 INTRODUCCIÓN AL PROCESADO DIGITAL DE IMÁGENES CON MATLAB

Es fundamental tener un mínimo conocimiento de Matlab y sus funciones básicas para poder utilizar cualquiera de las diferentes toolboxes especializadas de las que se dispone. En este apartado se explican las funciones y operaciones generales necesarias para iniciarse con Matlab.

4.3.1 El entorno de Matlab

En el entorno de Matlab aparecen 3 elementos principales:

- **Ventanas:** son de diversos tipos:
 - Las ventanas que se organizan en el escritorio y constituyen el núcleo del programa.
 - Figuras, editores de ficheros o variables: se abren y se cierran de forma frecuente. Son las que más aparecen en el procesado digital de imágenes.
 - Helps y demos: son ventanas específicas que hacen referencia a las demostraciones y a las demos.
- **Variables:** se tratan de objetos temporales que mientras la sesión este en curso son almacenadas en el Workspace, pero que cuando se cierra Matlab se borran.
- **Ficheros:** se tratan de objetos permanentes, que permiten guardar funciones o códigos directamente ejecutables en el programa. Entre ellos destacan los “.m”

4.3.2 El escritorio de Matlab

Al ejecutar el programa Matlab, aparece la interface de Matlab (*Figura 4-8*).

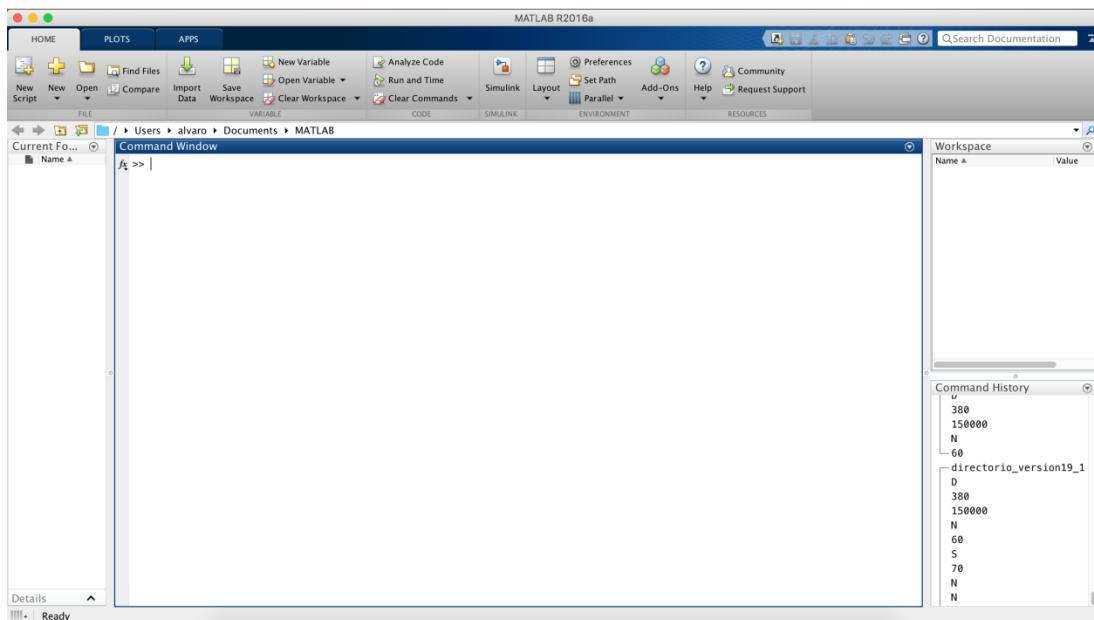


Figura 4-8: Interface de Matlab.

En la interface de Matlab se pueden encontrar distintas zonas, siendo las de mayor interés las siguientes:

- **Current Folder:** en ella aparecen los ficheros y carpetas que se encuentran en el directorio actual. Es el lugar donde se ejecutan los comandos dando la posibilidad de leer y escribir datos desde o en ese directorio.
- **Command Window:** se trata de la ventana principal, el lugar donde son tecleados los comandos de Matlab. Las instrucciones son tecleadas seguidas del prompt (>>) y ejecutadas pulsando la tecla Enter. Existe la posibilidad de volver a instrucciones tecleadas anteriormente sin necesidad de volver a escribirlas por medio de las flechas del teclado arriba y abajo.
- **Workspace:** es la ventana en la que aparecen las variables que han sido creadas durante la ejecución del programa, junto con el valor y el tipo del que se trata.
- **Command History:** ventana donde quedan almacenadas las instrucciones que han sido ejecutadas en Matlab en la sesión actual o en las anteriores y que están acompañadas por la fecha y la hora de la sesión.

Existen una serie de comandos que facilitan el manejo de Matlab:

- **clear all:** borra las variables almacenadas en el Workspace.
- **clc:** borra todos los comandos y respuestas escritas en el Command Window.
- **close all:** cierra todas las ventanas abiertas: figuras, aplicaciones, etc.
- **exit, quit:** cierra Matlab.
- **CTRL+C:** permite detener la ejecución de algún cálculo, gráfico o impresión sin salir del programa.
- **diary on/off:** graba la sesión ejecución en un “*.txt”.
- **hold on/off:** mantiene abierta una gráfica, lo que permite que los comandos ejecutados posteriormente se ejecuten en esa gráfica.
- **ver:** muestra la versión de Matlab y las *toolboxes* instaladas.
- **help “comando”:** aporta información sobre cualquier comando.

4.3.3 Variables y ficheros

Como ya se ha comentado anteriormente, las variables creadas durante la sesión en curso son almacenadas en el Workspace, que permite su modificación o su utilización en otros comandos. Existen dos formas de conocer el valor de una variable:

- Teclear su nombre en el Command Window.

- Ir al Workspace y hacer doble clic sobre el nombre de la variable.

Para poder utilizar las variables de una sesión anterior en la sesión actual, estas deben ser guardadas en un fichero Matlab.mat, lo que permita que sean cargadas posteriormente con el uso del comando “Load”. Además, con el comando diary se puede guardar el texto de la sesión en un “*.txt” determinado.

Mientras que las variables son la información temporal de Matlab, los ficheros son la información permanente que no se borra al cerrar la sesión. Existen dos tipos principales de ficheros:

- **Ficheros *.mat:** almacena datos que provienen de distintos formatos que con el uso del comando save y del comando load, son guardados y cargados.
- **Ficheros *.m:** contiene funciones con parámetros de entrada y salida o solamente comandos ejecutables.

Existe también la posibilidad de importar datos de otros ficheros (*.txt, *.xls, etc). Si el fichero se encuentra en el directorio actual se debe de hacer clic sobre el fichero y seleccionar la opción de importación o utilizar comandos con funciones específicas para la importación (importdata, fopen, fread, etc) mediante su escritura en la Command Window. Sin embargo, si los ficheros que se quieren importar se encuentran en otros lugares, se debe indicar a Matlab que busque en dichos lugares mediante el comando addpath.

Existen una serie de comandos que facilitan el manejo de variables y ficheros:

what: indica los ficheros que se encuentran en el directorio actual.

lookfor: permite la búsqueda de funciones o comandos relacionados con la palabra especificada.

size: mide el tamaño de una matriz en filas y columnas.

whos: muestra información adicional relativa a las variables actuales.

length: mide la longitud de un vector.

4.3.4 Comandos y matrices

Por medio de la línea de comandos se obtiene la manera más directa de interactuar con Matlab. Los comandos son tecleados en la Command Window seguidos del prompt y se

ejecutan con el Enter. Si una instrucción es demasiado larga y ocupa más de una línea, se escriben 3 puntos seguidos y se continúa en la línea siguiente. Si lo que se quiere es añadir comentarios se utiliza el símbolo “%”.

Si no se le asigna una variable a la instrucción, al ejecutarse el resultado será asignado a la variable “ans”. Si no se quiere que Matlab muestre el resultado en la Command Window se debe de añadir un “;” al final de la instrucción.

La base de Matlab es el trabajo con matrices numéricas rectangulares. Una manera sencilla de introducir matrices pequeñas es la siguiente (*Figura 4-9*):

- Los elementos deben estar escritos entre corchetes [].
- La separación de los elementos en columnas se lleva a cabo por medio de comas o espacios en blanco.
- Para pasar de una fila a la siguiente se utilizan “;”.

The screenshot shows the Matlab Command Window with the title bar "Command Window". In the window, the command `>> A=[1 2 3;4 5 6;7 8 9]` is entered and executed. The output shows the matrix A defined as:

```
A =  
1 2 3  
4 5 6  
7 8 9
```

The cursor is at the bottom of the window, indicated by the text `fx >>`.

Figura 4-9: Matriz en Matlab.

Las matrices pueden estar formadas por:

- **Un escalar:** matriz de un solo elemento.
 - `A=1`: define a A como un escalar de valor 1.
- **Un vector:** matriz de una sola dimensión que puede ser una fila o una columna un solo elemento.
 - `A = [1 2 3]`: define a A como un vector o array fila de tres elementos: $A(1)=1$, $A(2)=2$ y $A(3)=3$.
- **Un array:** matriz de dos dimensiones constituido por una serie de filas y columnas.

- $A=[1, 2, 3; 4, 5, 6]$: define a A como una matriz o array de dos dimensiones.

Uno de los operadores con un mayor empleo dentro de Matlab son los dos puntos “:”, debido a que dan la posibilidad de crear vectores, subarrays y determinar iteraciones:

- “ $j:k$ ”: vector formado por todos los valores entre j y k, ambos incluidos.
- “ $j:i:k$ ”: vector formado por los valores comprendidos entre j y k, ambos incluidos, y que están espaciados un valor i.
- $A(:,j)$: vector constituido por todas las filas y la columna j de la matriz A.
- $A(:,j:k)$: matriz formada por todas las filas y las columnas comprendidas entre j y k (ambas incluidas) pertenecientes a la matriz A.

Si se quiere trabajar con determinados elementos pertenecientes a las matrices se puede hacer de la siguiente manera:

- $A(1,1)$: se obtiene el elemento de la fila 1 y columna 1 perteneciente a la matriz A.
- $A(1:2,3)$: se obtienen las dos primeras filas de la matriz A y la tercera columna.
- $A(3,:)$: se obtienen todas las columnas pertenecientes a la fila 3.
- $B=A(1:2,2:3)$: se obtiene la matriz B, constituida por las dos primeras filas y las columnas 2 y 3 pertenecientes a la matriz A.

Las operaciones más básicas y los símbolos más utilizados con matrices y vectores son los siguientes:

- $+$: sumar.
- $-$: restar.
- $*$: multiplicar.
- $/$: dividir.
- $'$: trasponer.
- $^\wedge$: elevar una matriz a un número.

En determinadas ocasiones es necesario poner un punto antes de los signos y seguido de la matriz si se quiere que las operaciones se realicen componente a componente.

Existen una serie de comandos que pueden facilitar el manejo y la creación de matrices:

- **$zeros(N)$, $zeros(M,N)$** : matriz únicamente formada por ceros cuyas dimensiones son NxN o MxN de clase double.

- **zeros(size(A)):** matriz de ceros de las mismas dimensiones que la matriz A.
- **ones(N), ones(M,N):** matriz únicamente formada por unos cuyas dimensiones son NxN o MxN de clase double.
- **true(N), true(M,N):** matriz únicamente formada por unos cuyas dimensiones son NxN o MxN de clase logical.
- **false(N), false(M,N):** matriz únicamente formada por ceros cuyas dimensiones son NxN o MxN de clase logical.
- **magic(M):** matriz muy utilizada para probar resultados. Consiste en una matriz de MxM donde la suma a lo largo de cualquier fila, columna o diagonal principal da el mismo resultado.
- **rand(N), rand(M, N):** matriz de dimensiones NxN o MxN formada por números aleatorios comprendidos entre 0 Y 1.
- **randn(N), randn(M,N):** matriz de dimensiones NxN o MxN con sus valores distribuidos uniformemente con números de media 1 y varianza 0.

4.3.5 Representación de imágenes

Como ya se ha comentado anteriormente, el término imagen hace referencia a una función bidimensional de intensidad de luz $f(x, y)$ donde “x” e “y” hacen referencia a las coordenadas espaciales y el valor de f en cualquier punto (x, y) es proporcional a la intensidad de la imagen en ese mismo punto. Las imágenes en color en realidad son un conjunto de imágenes individuales, como en el típico espacio de trabajo RGB, que es una combinación de los componentes rojo, verde y azul. En base a esto, se puede decir que los procesos aplicados a las imágenes en escala de grises también pueden ser aplicado a las imágenes de color tratando cada componente de forma independiente.

Cuando se digitaliza una imagen, se obtiene una matriz $M \times N$ constituida por números reales, cuyas coordenadas han sido digitalizadas por muestreo y los valores de amplitud por cuantificación.

Mientras que en la Image Processing Toolbox, para indicar las filas y las columnas, se utiliza la notación (r, c) siendo el origen de coordenadas $(r, c) = (1,1)$, en la bibliografía del procesado digital de imágenes se utiliza la notación (x, y) con el origen de coordenadas $(x, y) = (0,0)$.

4.3.6 Entrada y salida de imágenes

4.3.6.1 Lectura de imágenes

La lectura de imágenes se lleva a cabo por medio de la línea de comandos de Matlab mediante el comando “imread”, añadiendo así la variable al Workspace:

```
Imagen = imread('nombre_imagen');
```

Si la imagen se encuentra en el directorio actual solo hay que sustituir ‘nombre_imagen’ por el nombre que tenga la imagen que se quiere leer. Mientras que, si la imagen no se encuentra en el directorio hay que sustituir ‘nombre_imagen’ por la ruta completa al archivo.

Con el comando “size”, se obtiene la dimensión de la imagen que se quiere analizar:

```
[r,c] = size (Imagen)
```

siendo r, el número de filas y c, el número de columnas.

Para obtener toda la información relacionada con la imagen, como son: sus características, sus dimensiones, sus bytes ocupados y la clase de archivo, se usa el comando whos:

```
whos Image
```

4.3.6.2 Visualización de imágenes

La visualización de imágenes se lleva a cabo por medio del comando “imshow”. Cuando se ejecuta este comando se abre la imagen definida en una nueva ventana, pero sin generar una salida de datos en el Command Window. Si se ejecuta la función de nuevo, sin haber especificado una nueva figura, comando “figure”, esta nueva imagen sustituye a la anterior.

Las sintaxis más comunes son las siguientes:

- **imshow(Imagen, G):** siendo “Imagen”, la imagen a mostrar y “G” el número de niveles de intensidad que son utilizados para mostrar la imagen, 256 es el valor por defecto.
- **imshow(Imagen, [low high]):** siendo “Imagen” la imagen a mostrar y donde todos los valores que son menor o igual que low se muestren de color negro y los que son iguales o mayores que high blancos. Los valores que se encuentren entre low y high son mostrados por intensidad en base al número de niveles por defecto.

4.3.6.3 Escritura de imágenes

La escritura de imágenes se lleva a cabo por medio del comando “imwrite”. Las imágenes siempre son guardadas en el directorio actual de trabajo a no ser que se especifique otro directorio. Las funciones más habituales son:

imwrite(f, 'nombre_archivo')

imwrite(f, 'nombre_archivo', 'tipo_imagen')

imwrite(f, 'nombre_archivo.extensión')

En el caso de archivos JPG, existe la posibilidad de definir la calidad del archivo final mediante la variable **q**, que debe de tener un valor comprendido entre 0 y 100:

imwrite(f, 'nombre_archivo.jpg', 'quality', q)

Existen dos maneras para guardar las imágenes que han sido generadas con Matlab:

- **Menú:** File – Save as... y se elige el tipo de formato que se le quiere dar.
- **Command Window:** tecleando la función print:
 - **print -f1 -dtiff -r300 prueba**
 - f1: el número de figura.
 - dtiff: el formato “tif”.
 - r300: resolución “300”.
 - prueba: nombre que va a recibir el archivo.

4.3.7 Tipos de datos y de imágenes

4.3.7.1 Tipos de datos

Los tipos de datos [11] con los que trabaja Matlab pueden ser agrupados en tres grandes grupos: numéricos, de caracteres y lógicos. La *Tabla 1* define cada uno de estos grupos:

NOMBRE	DESCRIPCIÓN
<i>double</i>	Números de precisión doble entre -10^{308} y 10^{308} . Si es una imagen con valores escalados el rango es [0, 1] (8 bytes por elemento)
<i>uint8</i>	Enteros sin signo de 8-bits en el rango [0, 255] (1 byte por elemento)
<i>uint16</i>	Enteros sin signo de 16-bits en el rango [0, 65535] (2 byte por elemento)
<i>uint32</i>	Enteros sin signo de 8 bits en el rango [0, 4294967295] (4 byte por elemento)
<i>int8</i>	Enteros sin signo de 8 bits en el rango [-128, 127] (1 byte por elemento)
<i>int16</i>	Enteros sin signo de 8 bits en el rango [-32768, 32767] (2 byte por elemento)
<i>int32</i>	Enteros sin signo de 8 bits en el rango [-2147483648, 2147483647] (4 byte por elemento)
<i>single</i>	Número de precisión simple entre -10^{38} y 10^{38} (4 bytes por elemento)
<i>char</i>	Caracteres (2 bytes por elemento)
<i>logical</i>	Valores de 0 a 1 (1 byte por elemento)

Tabla 1: Tipos de datos con los que trabaja Matlab.

Se puede realizar una conversión de datos por medio de las funciones de la *Tabla 2* que aporta la Image Processing Toolbox:

FUNCIÓN	CLASE DE ENTRADA	CLASE DE SALIDA
<i>im2uint8</i>	logical, uint8, uint16, double	uint8
<i>Im2uint16</i>	logical, uint8, uint16, double	uint16
<i>mat2gray</i>	double	double escalada (rango [0, 1])
<i>im2double</i>	logical, uint8, uint16, double	double
<i>im2bw</i>	uint8, uint16, double	logical

Tabla 2: Funciones para la conversión de datos.

4.3.7.2 Tipos de imágenes

Existen 4 tipos distintos de imágenes [11] [12] que son soportadas por la toolbox, de las cuales las binarias y las de intensidad son las que más uso tienen dentro del procesado digital de imágenes monocromáticas. Los cuatro tipos de imágenes son las siguientes:

- **Imágenes de intensidad:** se trata de matrices de datos cuyos valores han sido escalados para representar una escala determinada de intensidad.
- **Imágenes binarias:** se trata de arrays compuestos por ceros y unos de clase logical. Si se quiere convertir un array numérico en uno binario se puede utilizar la función “logical”. Mediante la función “islogical” se puede comprobar si un array es binario.
- **Imágenes RGB (Red Green Blue):** se trata de arrays de pixeles de color con un tamaño de MxNx3, en donde cada pixel está constituido por una tripla de valores correspondientes a los componentes rojo, verde y azul de la imagen en una determinada posición. El rango de valores viene determinado por el tipo de datos de los componentes.

La profundidad de color en una imagen RGB viene definidaa por la cantidad de bits que han sido utilizados para representar los valores de los pixeles en cada componente de la imagen. Con la ayuda de la función “cat”, se puede crear una imagen RGB a partir de sus componentes por separado. En cambio, si se quieren extraer los componentes de una imagen RGB se debe de utilizar la siguiente programación:

$$R = RGB(:, :, ,1);$$

$$G = RGB(:, :, ,2);$$

$$B = RGB(:, :, ,3);$$

- **Imágenes indexadas:** están constituidas por dos componentes:
 - Matriz de datos X: está formada por números enteros.
 - Matriz Map: formada por la paleta de color RGB.

La matriz del mapa consiste en un array de Mx3 de clase double que contiene valores comprendidos entre 0 y 1 y donde la longitud de M coincide con el número de colores que define.

La Image Processing Toolbox dispone de una serie de funciones, las cuales permiten convertir un tipo de imagen en otra, siendo las más habituales las de la *Tabla 3*:

FUNCIÓN	IMAGEN DE ENTRADA	IMAGEN DE SALIDA
<i>dither</i>	Imagen RGB	Imagen indexada (mediante trámado)
<i>grayslice</i>	Imagen en escala de grises	Imagen indexada (mediante umbralización multinivel)
<i>gray2ind</i>	Imagen en escala de grises	Imagen indexada
<i>ind2gray</i>	Imagen indexada	Imagen en escala de grises
<i>rgb2ind</i>	Imagen RGB	Imagen indexada
<i>ind2rgb</i>	Imagen indexada	Imagen RGB
<i>rgb2gray</i>	Imagen RGB	Imagen en escala de grises

Tabla 3: Funciones para la conversión de imágenes.

4.4 IMAGE PROCESSING TOOLBOX

La Image Processing Toolbox [10] [17] permite el procesamiento, análisis y visualización de imágenes, así como llevar a cabo el desarrollo de algoritmos, debido a que dispone de un conjunto completo de algoritmos estándar de referencia y apps de flujo de trabajo. Además, permite realizar segmentación de imágenes, mejora de imágenes, reducción de ruido, transformaciones geométricas, registro de imágenes y procesamiento de imágenes 3D.

Las apps pertenecientes a la Image Processing Toolbox permiten que los flujos de trabajo habituales para el procesamiento de imágenes se encuentren automatizados. Se puede segmentar datos de las imágenes, comparar técnicas de registro de imágenes y procesar conjuntos de datos extensos de forma interactiva por lotes. Las apps y las funciones destinadas a la visualización dan la posibilidad de llevar a cabo exploraciones de imágenes, volúmenes en 3D y videos, ajustar el contraste, crear histogramas y manipular regiones de interés (ROIs).

La Image Processing Toolbox, con el uso de procesadores multinúcleo y GPUs, es capaz de acelerar los algoritmos. Una gran cantidad de funciones pertenecientes a esta toolbox son

Aplicaciones Técnicas de Visión Artificial a Balística de Efectos

capaces de soportar la generación de código C/C++ para el desarrollo de sistemas de visión embebidos.

5 ENSAYOS REALIZADOS

Fueron realizados 6 ensayos con 3 tipos de proyectiles [15] distintos, los cuales fueron disparados contra gelatina balística sin hueso y grabados mediante cámaras de alta velocidad para su posterior tratamiento. Cada modelo de proyectil fue disparado dos veces, en ninguno de los casos el mismo modelo de proyectil fue disparado al mismo bloque de gelatina balística. A continuación se detallan los elementos utilizados en los ensayos:

5.1 PROYECTILES

Los proyectiles utilizados han sido los siguientes:

5.1.1 FULL METAL JACKET (FMJ)

Hace referencia al tipo de revestimiento usado, aleación de cobre o de acero, cuyo objetivo es el de reducir la cantidad de residuos de plomo que quedan en el cañón una vez realizado el disparo.



Figura 5-1: Proyectil Full Metal Jacket.

Ventajas

- Es menos contaminante que una bala sin encamisado.
- El proyectil no se deforma tanto al impactar.
- Tiene una mayor penetración.

Inconvenientes

- Menor expansión en los tejidos blandos.
- Rebotes contra objetos duros.

5.1.1.1 Ensayo 1

En el primer ensayo, se ha realizado una grabación con una duración de 9 minutos y 35 segundos la cual ha sido cortada con la ayuda del programa Photron FASTCAM Viewer, y se han obtenido 523 frames, lo que equivale a los frames comprendidos entre la penetración del

proyectil en la gelatina balística hasta cuando la gelatina deja de expandirse. Estos frames posteriormente se han analizado con el programa creado en Matlab.

5.1.1.2 Ensayo 2

En el segundo ensayo, se ha realizado una grabación con una duración de 14 minutos y 29 segundos la cual ha sido cortada con la ayuda del programa Photron FASTCAM Viewer, y se han obtenido 523 frames, lo que equivale a los frames comprendidos entre la penetración del proyectil en la gelatina balística hasta cuando la gelatina deja de expandirse. Estos frames posteriormente se han analizado con el programa creado en Matlab.

5.1.2 PUNTA BLANDA (SP)

Se trata de una bala con el frente descubierto, también conocida como "semi-encamisada".

Los proyectiles SP dejan una parte expuesta a diferencia de las FMJ que se encuentran cerradas por completo, lo que hace que a la hora del impacto se expandan dando lugar a un área de superficie mayor.



Figura 5-2: Proyectil Punta Blanca.

Ventajas

- Expansión más lenta que las de punta hueca.
- Mayor velocidad de salida que las de punta hueca.

Inconvenientes

- Posibles atasques en la rampa del cañón.

5.1.2.1 Ensayo 1

En el primer ensayo, se ha realizado una grabación con una duración de 2 minutos y 46 segundos la cual ha sido cortada con la ayuda del programa Photron FASTCAM Viewer, y se han obtenido 403 frames, lo que equivale a los frames comprendidos entre la penetración del

proyectil en la gelatina balística hasta cuando la gelatina deja de expandirse. Estos frames posteriormente se han analizado con el programa creado en Matlab.

5.1.2.2 Ensayo 2

En el segundo ensayo, se ha realizado una grabación con una duración de 9 minutos y 48 segundos la cual ha sido cortada con la ayuda del programa Photron FASTCAM Viewer y se han obtenido 540 frames, lo que equivale a los frames comprendidos entre la penetración del proyectil en la gelatina balística hasta cuando la gelatina deja de expandirse. Estos frames posteriormente se han analizado con el programa creado en Matlab.

5.1.3 PUNTA HUECA (HP)

Tienen un centro ahuecado. Cuando este tipo de proyectil alcanza su blanco el hueco producido hace que la bala se deforme y expanda lo que hace que sea capaz de empujar una superficie más grande de tejido.



Figura 5-3: Proyectil Punta Hueca.

Ventajas

- Orificios generados de gran tamaño en blancos blandos.
- Insignificantes rebotes en blancos duros.

Inconvenientes

- Se produce una disminución en la penetración en el acero y el hormigón.
- Si impacta contra un objetivo no deseado, el daño generado es enorme.

5.1.3.1 Ensayo 1

En el primer ensayo, se ha realizado una grabación con una duración de 4 minutos y 33 segundos la cual ha sido cortada con la ayuda del programa Photron FASTCAM Viewer y se han obtenido 631 frames, lo que equivale a los frames comprendidos entre la penetración del

proyectil en la gelatina balística hasta cuando la gelatina deja de expandirse. Estos frames posteriormente se han analizado con el programa creado en Matlab.

5.1.3.2 Ensayo 2

En el segundo ensayo, se ha realizado una grabación con una duración de 20 minutos y 40 segundos la cual ha sido cortada con la ayuda del programa Photron FASTCAM Viewer y se han obtenido 718 frames, lo que equivale a los frames comprendidos entre la penetración del proyectil en la gelatina balística hasta cuando la gelatina deja de expandirse. Estos frames posteriormente se han analizado con el programa creado en Matlab.

5.2 GELATINA BALÍSTICA

La gelatina balística es uno de los materiales sustitutivos utilizados para el estudio de la balística de efectos. Este producto permite el estudio y análisis de un gran número de aplicaciones que tienen una estrecha relación con el ámbito policial, militar o forense, como por ejemplo el diseño de chalecos antibalas.

La principal ventaja de este producto es que simula, de una manera muy aproximada, la viscosidad y la densidad del tejido humano por lo que permite hacerse una idea de los efectos que produce el impacto de un proyectil o de un cuchillo sobre una persona. Además, permite una reducción de costes de las pruebas realizadas, ya que este compuesto es más económico que otros materiales.



Figura 5-4: Gelatina Balística [24].

5.3 CÁMARA

Para la grabación de los ensayos se han utilizado 2 cámaras digitales monocromo de alta velocidad, *PHOTRON FASTCAM SA-Z* (*Figura 5-5*), que son capaces de captar hasta 2.1 millones de imágenes por segundo y de capturar secuencias de imágenes completas del proceso de avance del proyectil.



Figura 5-5: Cámara Photron Fastcam SA-Z

5.4 ILUMINACIÓN

Para los ensayos se ha utilizado un sistema de focos HV-VISION de tecnología LED (*Figura 5-6*). La sincronización de los focos con el ensayo se lleva a cabo mediante un trigger.



Figura 5-6: Iluminación.

6 MANUAL DE FUNCIONAMIENTO

Este apartado sirve para dar a conocer como poder poner en marcha este proyecto. Para ello es necesario disponer de los programas Photron FASTCAM Viewer para obtener los frames que interesan de los videos grabados y Matlab para ejecutar el programa creado “*analisis_proyectil*”.

En este capítulo, se utiliza el *proyectil fmj* para explicar el funcionamiento de los programas, siendo el procedimiento para el resto de ensayos exactamente el mismo. (En los ANEXOS II y III, se presentan los resultados obtenidos para los *projectiles sp y hp*).

6.1 PHOTRON FASTCAM VIEWER

Este programa puede ser encontrado en internet en el siguiente enlace: <http://photon.com/support/downloads/>. Se trata de un programa gratuito que permite cortar los videos y obtener imágenes de ellos. A continuación, se explican los pasos que hay que realizar para poder cortar el video y obtener las imágenes para poder analizarlas posteriormente.

1. Abrir el programa.

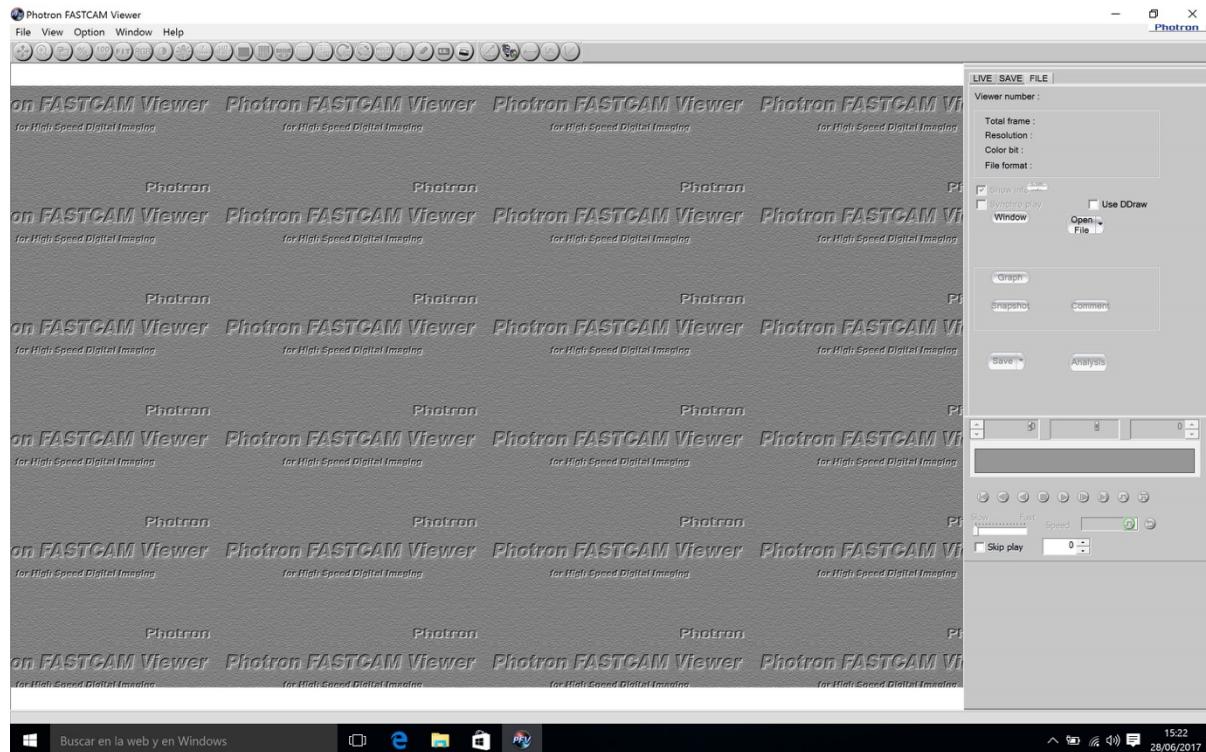


Figura 6-1: Interface Photron Fastcam Viewer.

2. Open file.
3. Buscar la carpeta en la que se encuentre el fichero.
4. Seleccionar el archivo *.cih, que nos aporta más información.

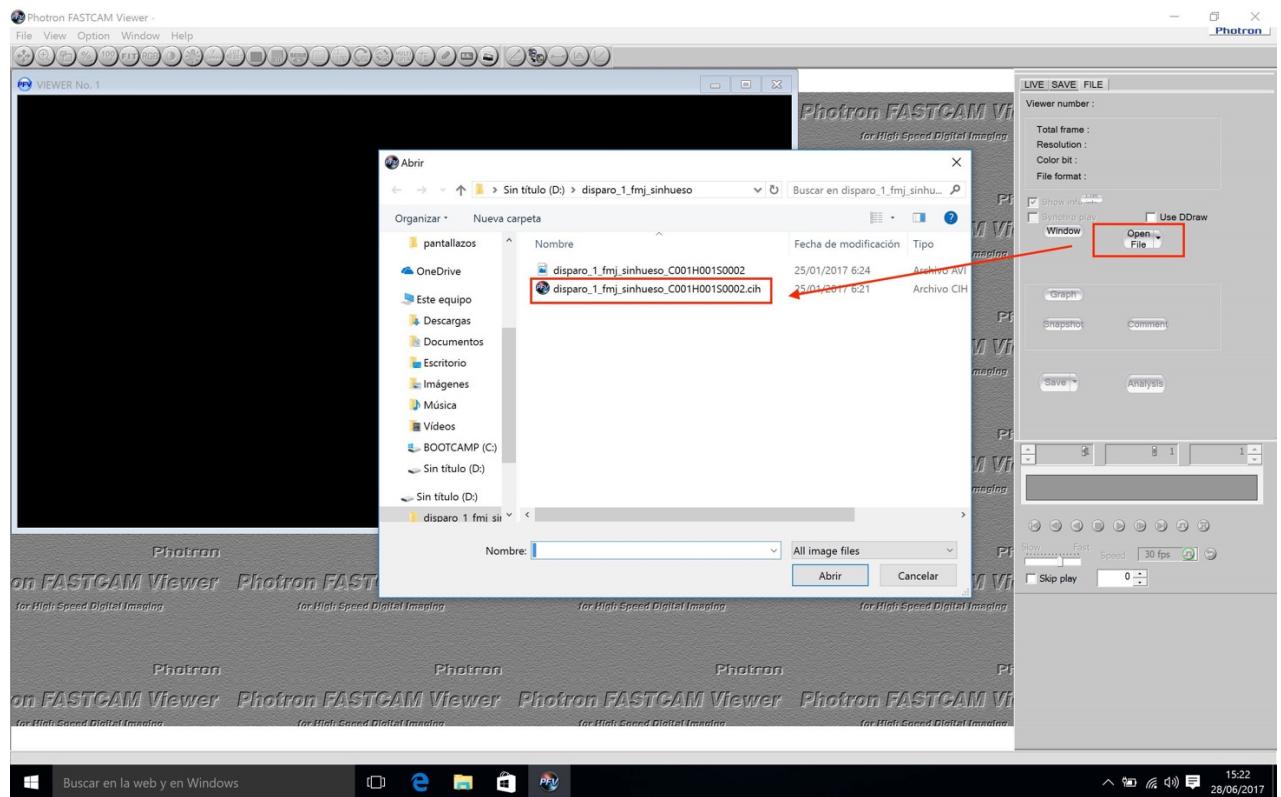


Figura 6-2: Selección del archivo.

Aplicaciones Técnicas de Visión Artificial a Balística de Efectos

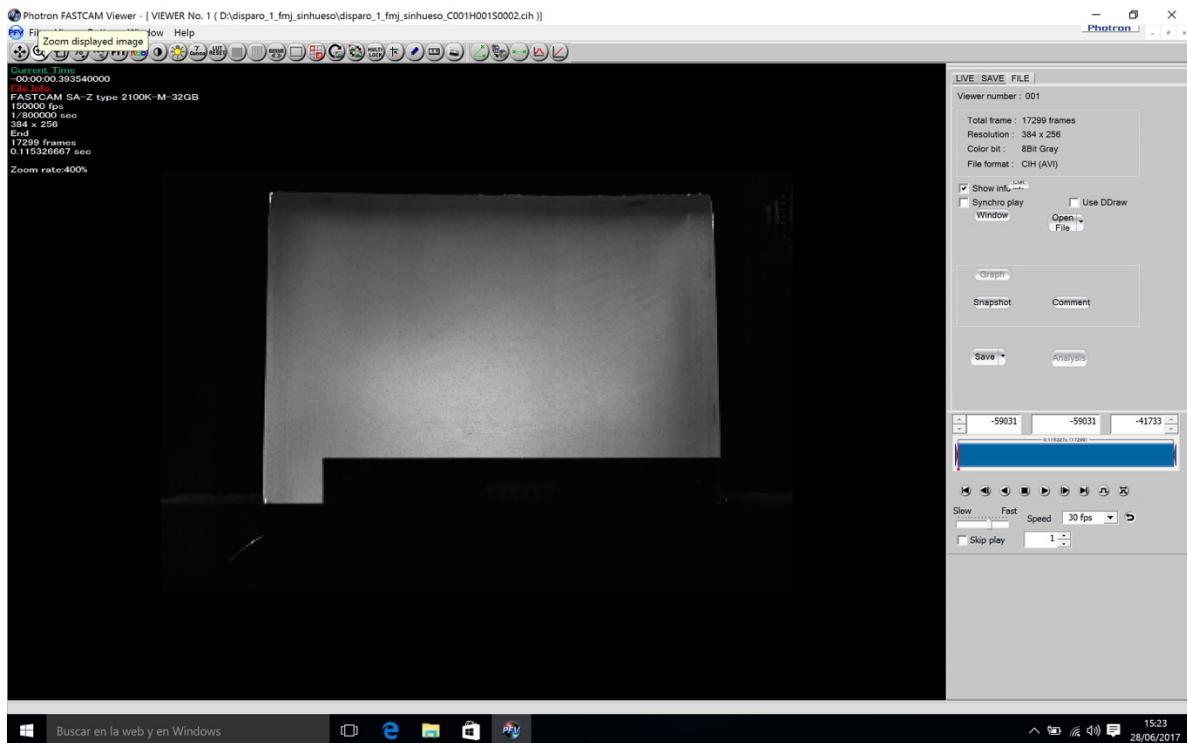


Figura 6-3: Interface del programa con el archivo abierto.

El fichero *.cih aporta una mayor información que el fichero *.avi, aunque con los dos se consigue el mismo resultado.

```
Current Time  
-00:00:00.393540000  
File Info.  
FASTCAM SA-Z type 2100K-M-32GB  
150000 fps  
1/800000 sec  
384 x 256  
End  
17299 frames  
0.115326667 sec  
  
Zoom rate:400%
```

Figura 6-4: Información aportada por el archivo *.cih.

5. Pulsar el botón “play” y tomar nota del frame, en el cual la bala penetra en la gelatina y el frame en el cual el orificio originado por la bala deja de expandirse.
6. En la Figura 6-5, se marca con un recuadro rojo el lugar donde deben ser introducidos los valores de ambos frames.

Aplicaciones Técnicas de Visión Artificial a Balística de Efectos

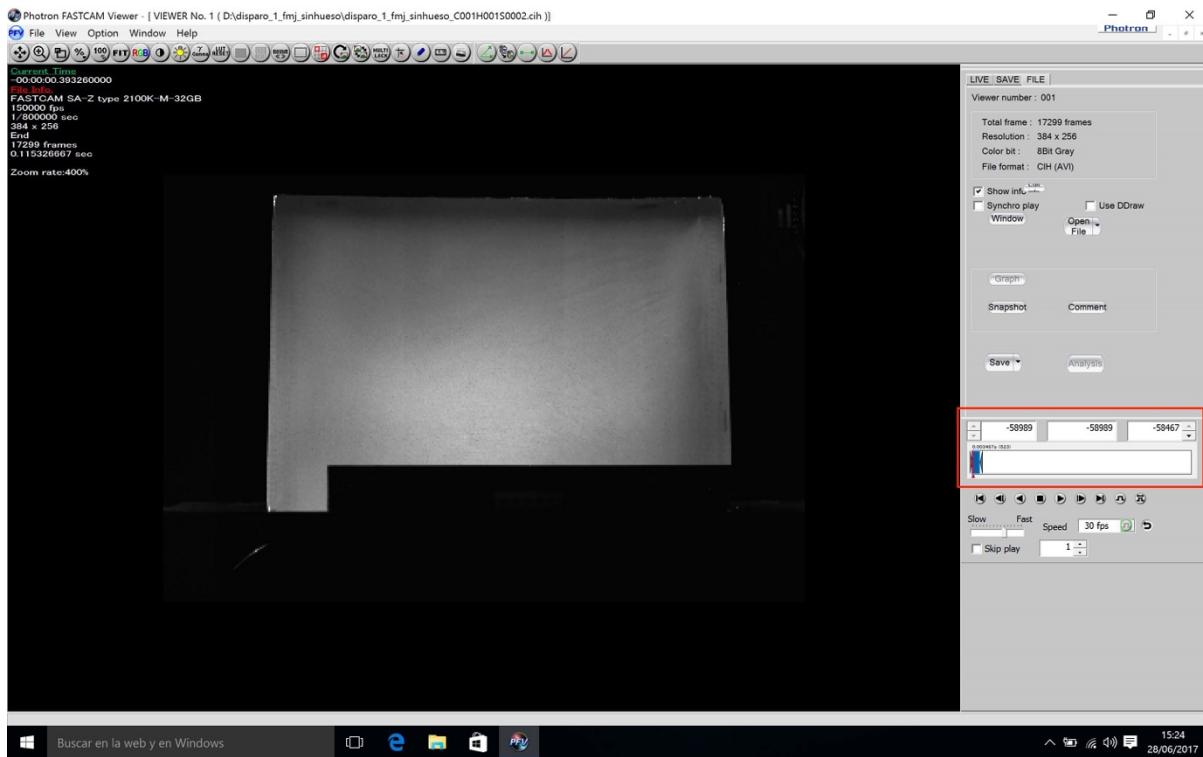


Figura 6-5: Introducción frames penetración y fin de expansión.

7. Save – Se abre una pantalla donde aparece la carpeta y el archivo con el que se ha estado trabajando.
8. Elegir el tipo de fichero (En este caso *.jpeg).

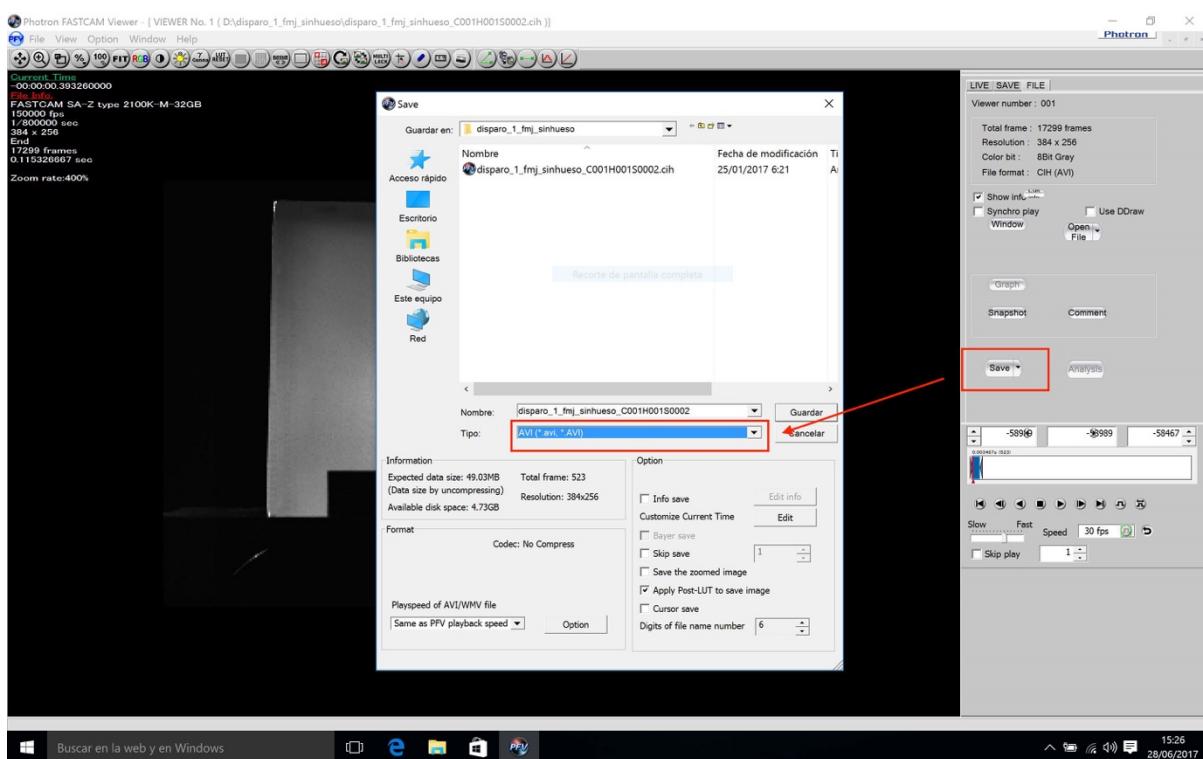


Figura 6-6: Guardas los frames.

Aplicaciones Técnicas de Visión Artificial a Balística de Efectos

9. Elegir la calidad de las imágenes, marcando el valor de 100%.

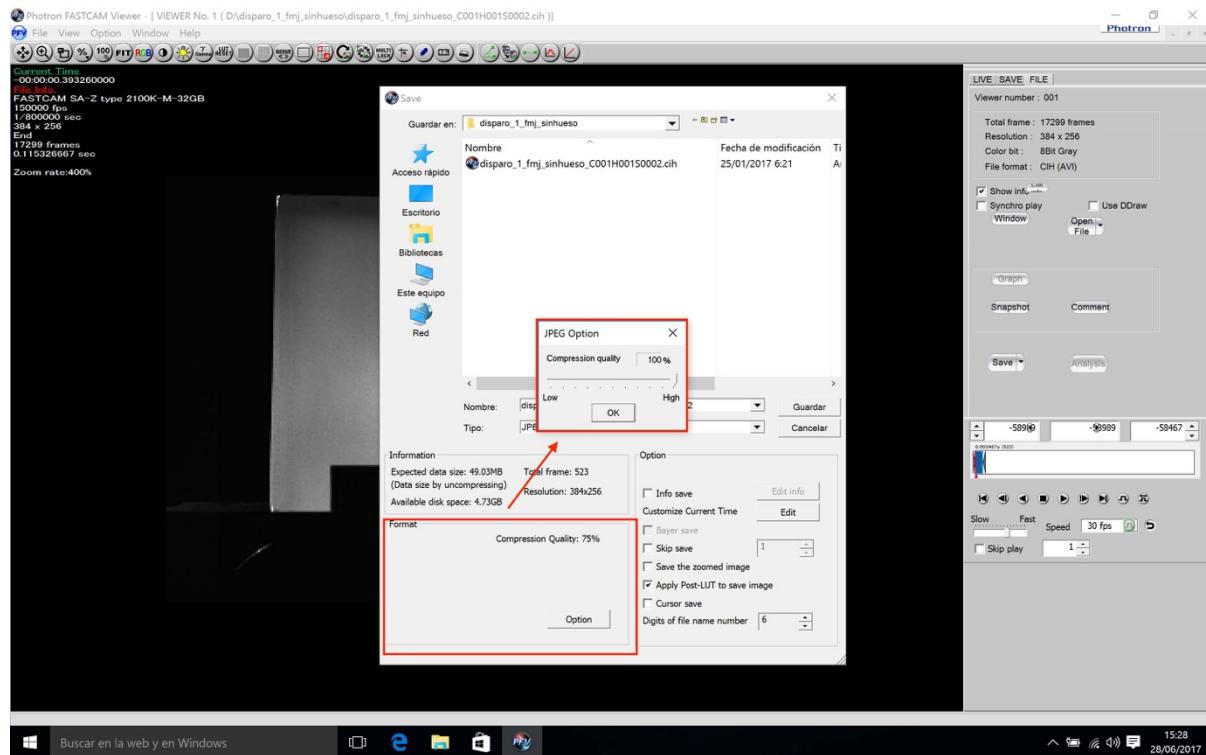


Figura 6-7: Selección de calidad de las imágenes.

10. Pulsar el botón guardar.

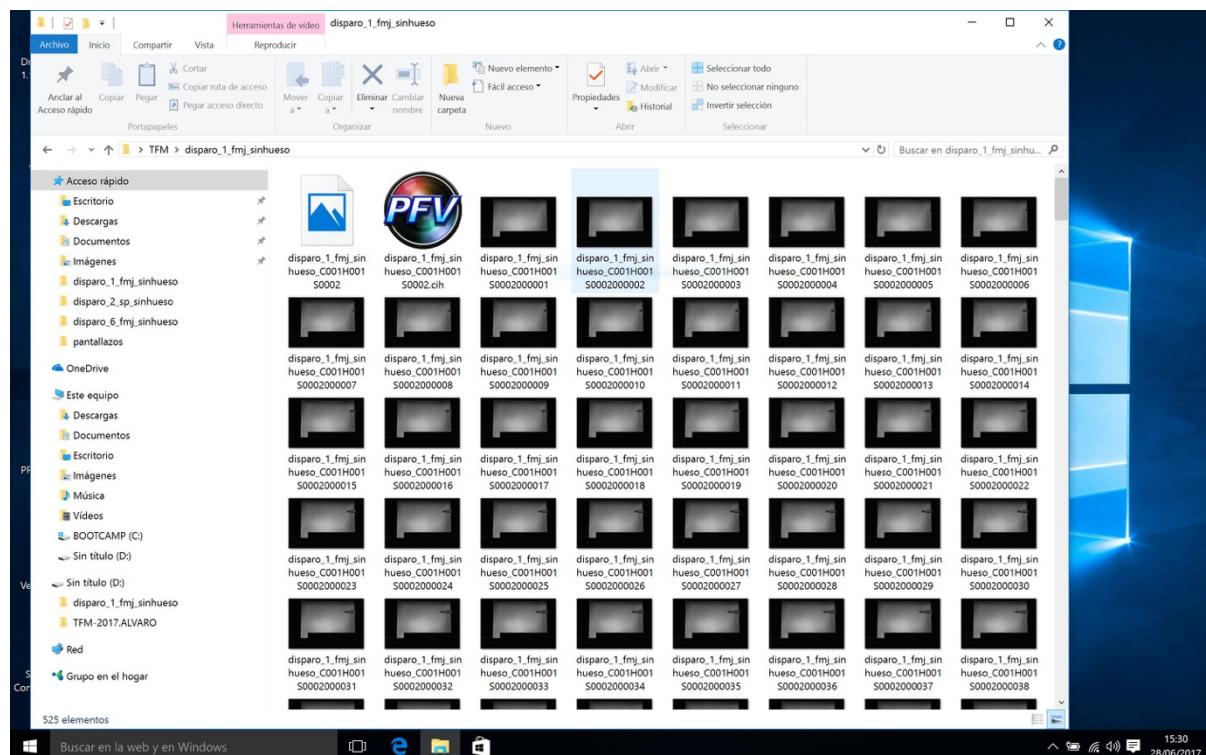


Figura 6-8: Imágenes en la carpeta original.

6.2 MATLAB

A continuación, se explican los pasos que hay que realizar para tratar las imágenes que han sido obtenidas del programa Photron FASTCAM Viewer y de esta manera obtener la posición, la velocidad, la aceleración y la energía transmitidas por el proyectil y el área y volumen generados por este en el cuerpo para posteriormente ser grabados en un Excel.

Este manual ha sido realizado mediante un proyectil fmj y en él se muestran las gráficas para este proyectil. En los Anexos 11.2 y 11.3 se muestran las gráficas generadas por un proyectil HP y SP respectivamente.

1. Abrir el programa Matlab.

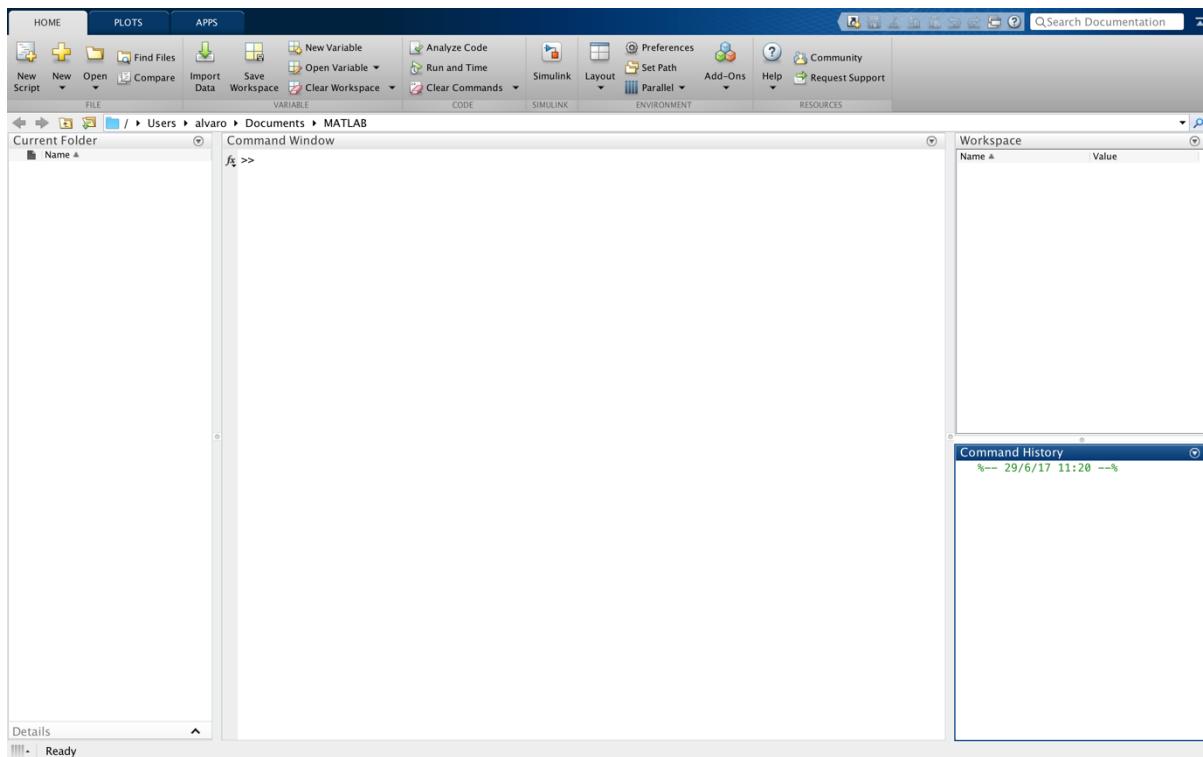


Figura 6-9: Interface Matlab

2. Buscar el directorio donde se encuentre el programa.

Aplicaciones Técnicas de Visión Artificial a Balística de Efectos

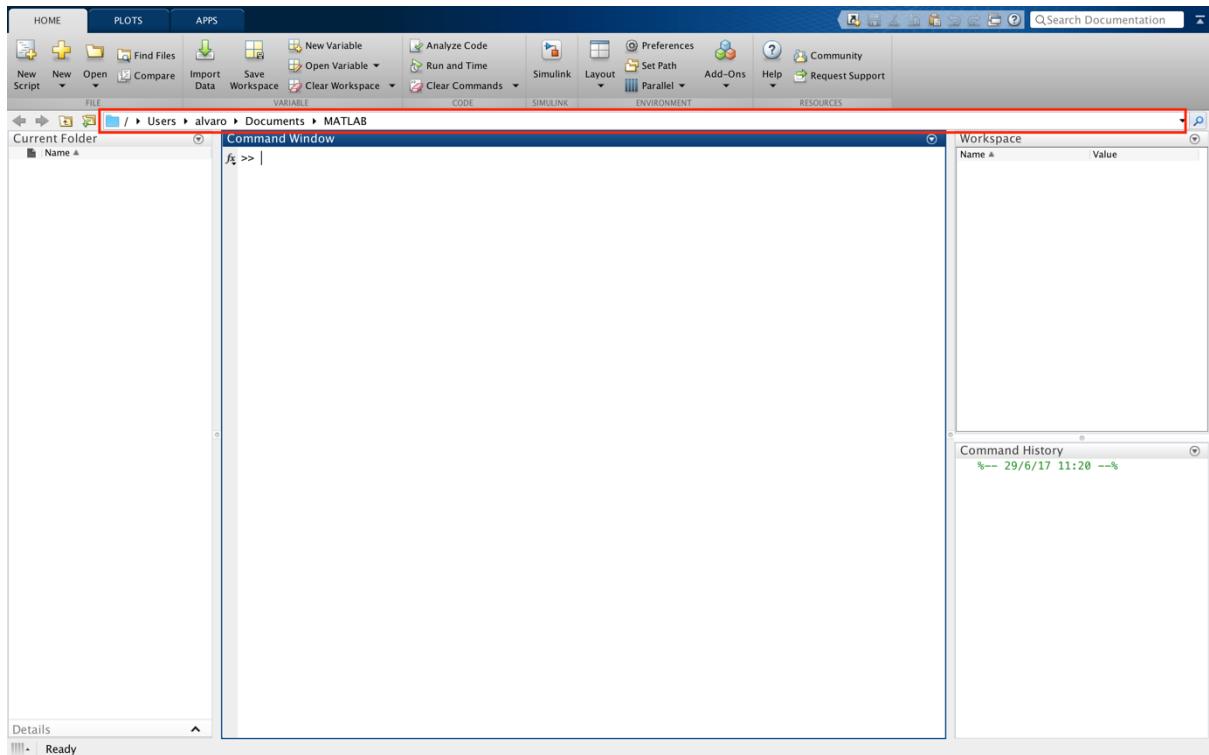


Figura 6-10: Guardas los frames.

3. Teclear el nombre de la aplicación creada “análisis_proyectil” en el Command Window.

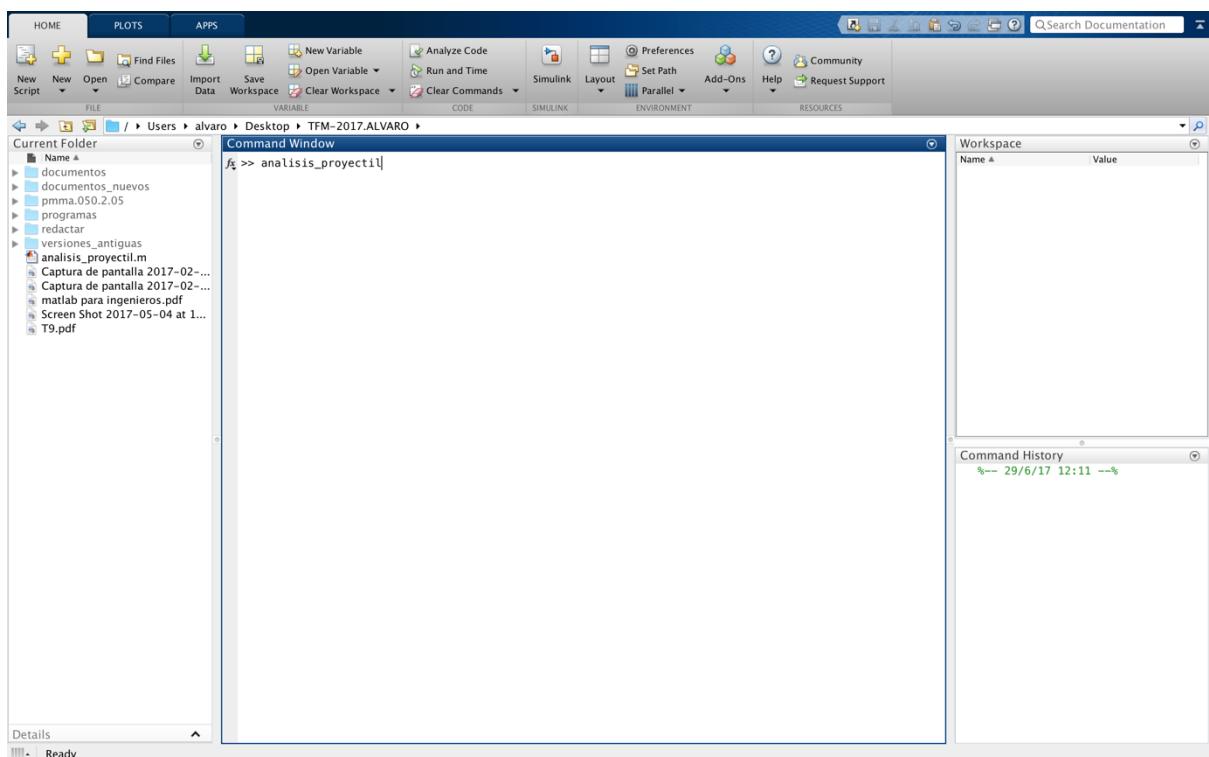


Figura 6-11: Llamada al programa desde el Command Window.

Aplicaciones Técnicas de Visión Artificial a Balística de Efectos

4. Se abre una ventana emergente para localizar la carpeta en la cual están guardadas las imágenes.

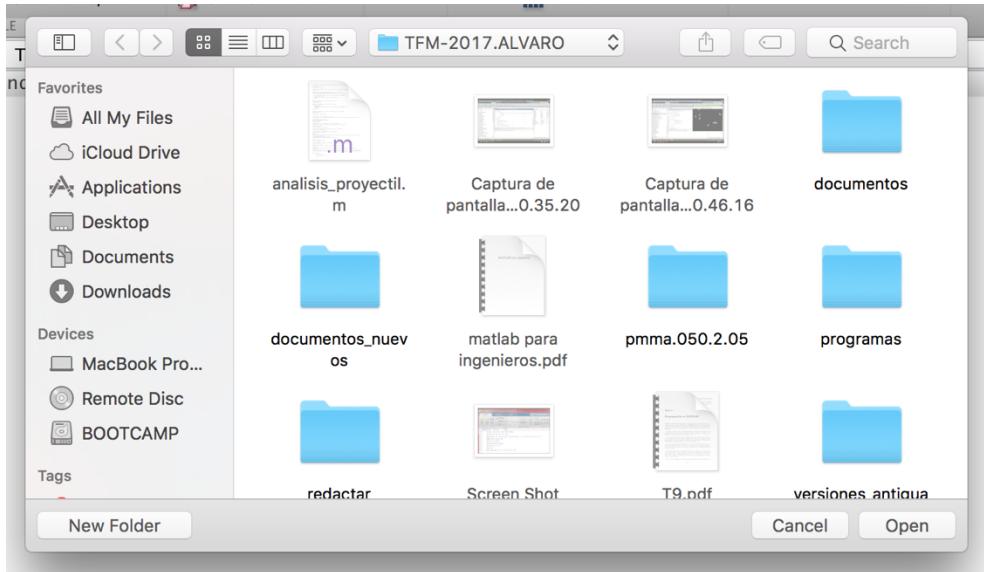


Figura 6-12: Ventana para localizar carpeta.

5. Indicar lado por el que entra el proyectil en la gelatina.

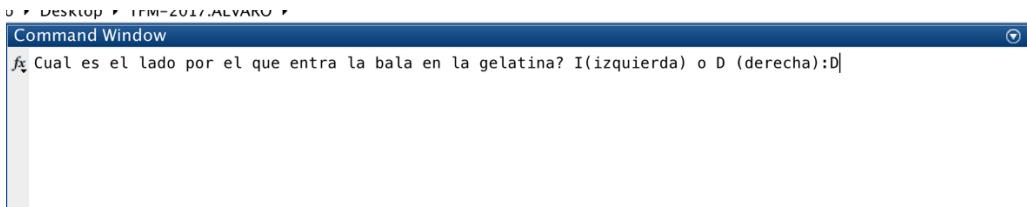


Figura 6-13: Indicar lado.

6. Determinar la masa del proyectil.

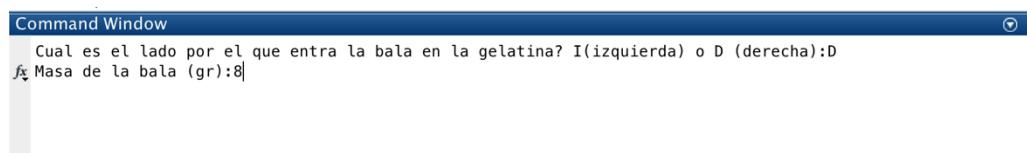


Figura 6-14: Determinar la masa de la bala.

7. Determinar el tamaño de la gelatina en mm.

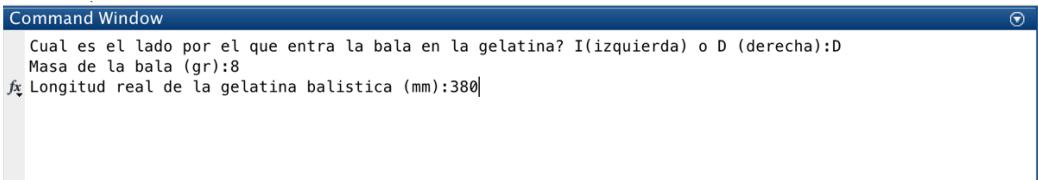


Figura 6-15: Determinar el tamaño de la gelatina balística.

8. Determinar los frames por segundo (Se obtiene del programa Photron FASTCAM Viewer).

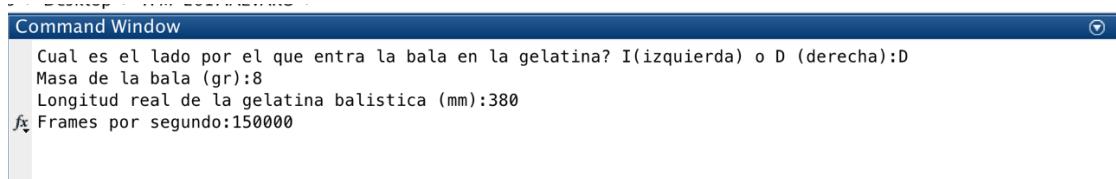


Figura 6-16: Determinar los frames por segundo.

9. Aparece una ventana emergente con una imagen en la cual se realiza el primer corte.

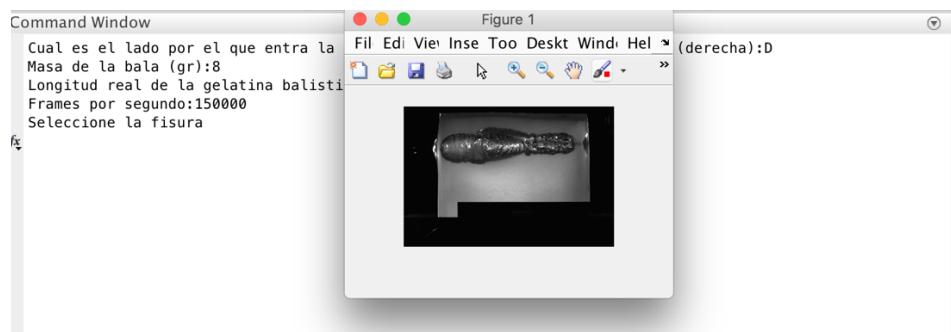


Figura 6-17: Ventana emergente para realizar primer corte.

10. Maximizar ventana.

11. Marcar con un cuadro la fisura.

12. Click derecho y "Crop Image".

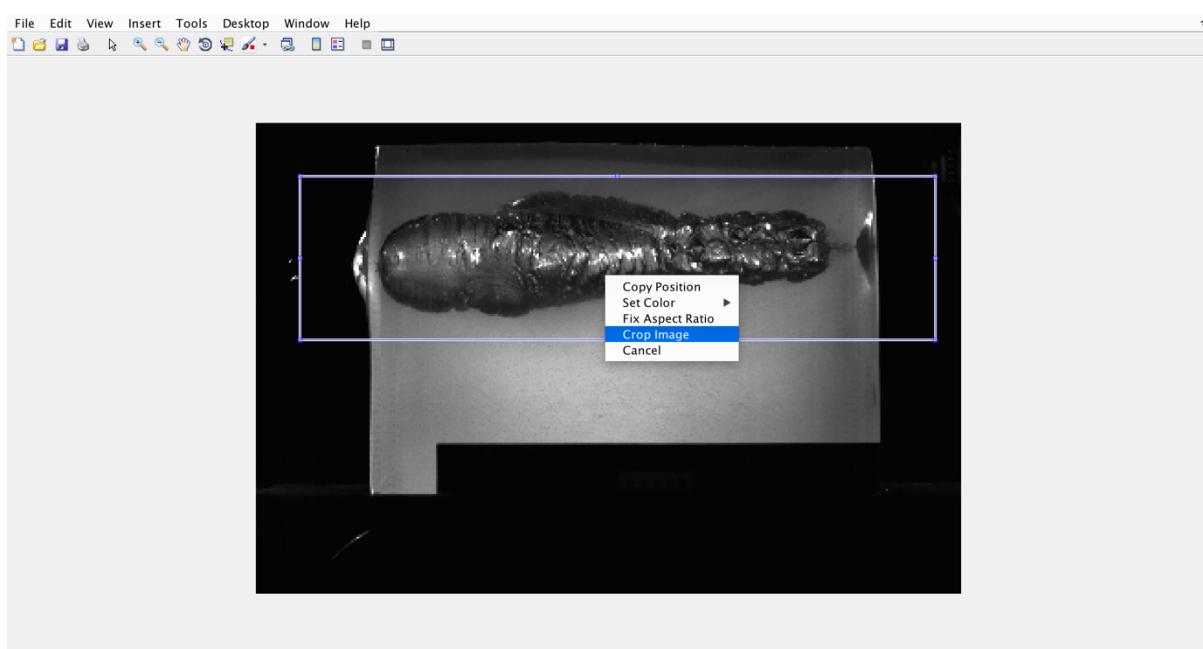


Figura 6-18: Cortar Imagen.

Aplicaciones Técnicas de Visión Artificial a Balística de Efectos

13. Si el corte no ha sido bien realizado, existe la posibilidad de volverlo a realizar tecleando "S".

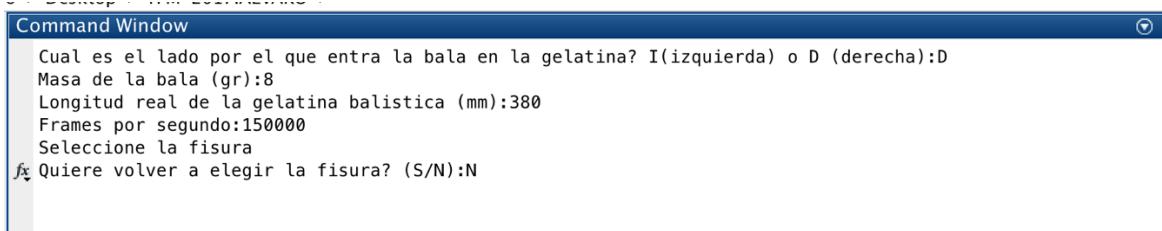


Figura 6-19: Indicar si se quiere realizar de nuevo el corte.

14. Definir el valor a partir del cual se binariza la imagen.

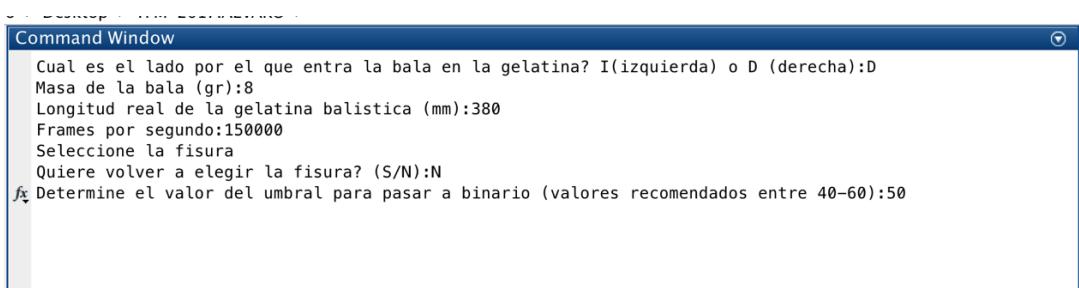


Figura 6-20: Definir umbral.

15. El programa pregunta si el valor elegido es correcto o si se quiere probar con otro valor.

Recomendaciones de valores en función del proyectil (valores orientativos).

- a. Proyectil fmj: 50.
- b. Proyectil sp: 65.
- c. Proyectil hp: 65.

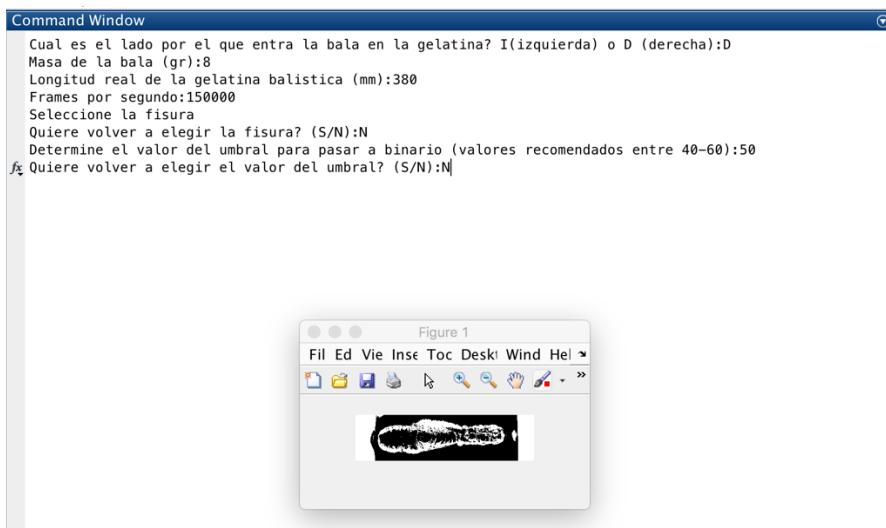


Figura 6-21: Binarizado imagen.

16. Ventana emergente para realizar un segundo corte a la imagen.

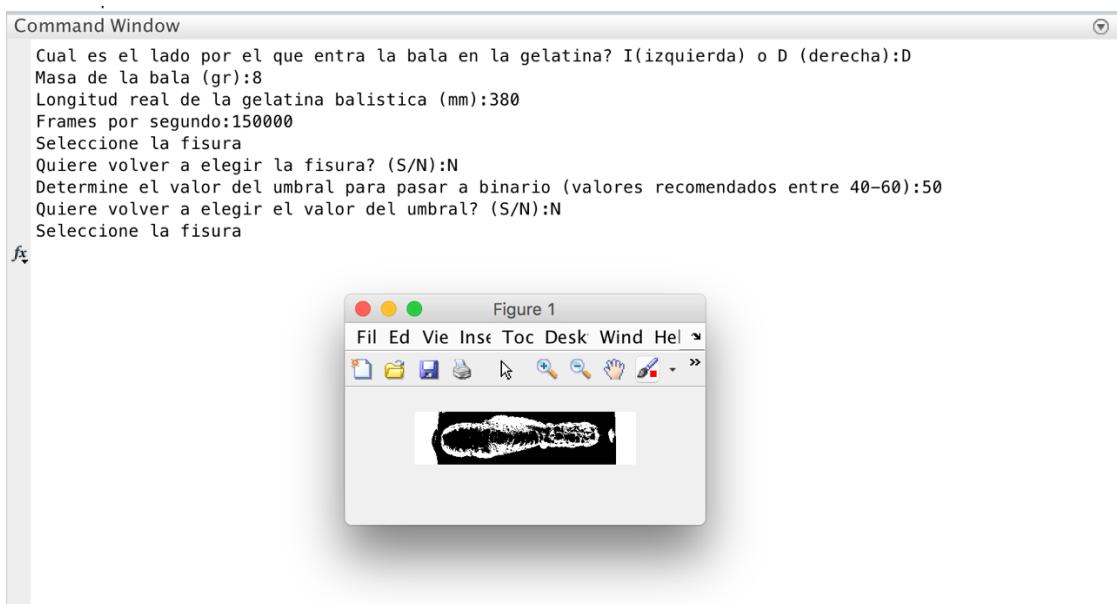


Figura 6-22: Ventana emergente para realizar segundo corte.

17. Maximizar ventana.

18. Marcar con un cuadro la fisura.

19. Click derecho y “Crop Image”.

- Es importante cortar la imagen de tal manera que todos los pixeles, tanto desde arriba como desde abajo, sean negros hasta encontrar la fisura.*
- Es importante dejar espacio en el lado por el que sale el proyectil.*

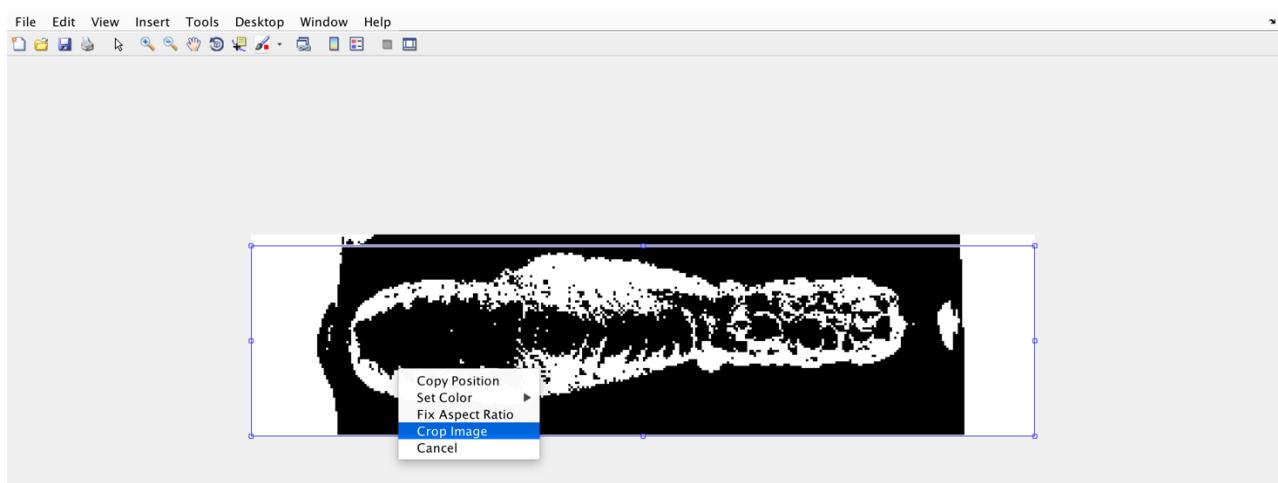


Figura 6-23: Segundo corte.

- 20. Si el corte no ha sido bien realizado, existe la posibilidad de volverlo a realizar tecleando “S.”**

Aplicaciones Técnicas de Visión Artificial a Balística de Efectos

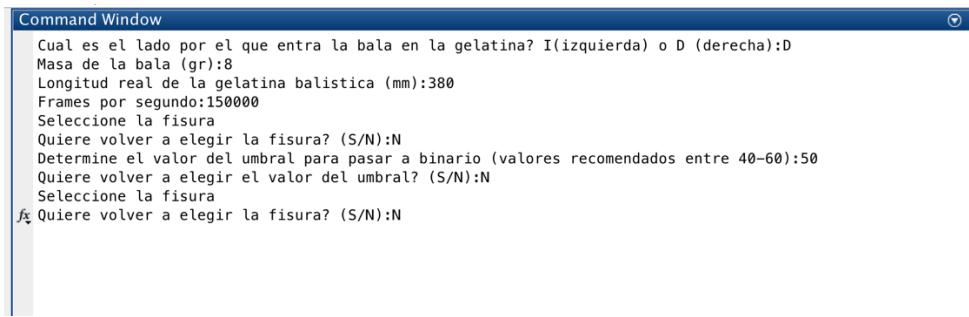


Figura 6-24: Indicar si se quiere realizar de nuevo el corte.

21. Comienza de forma automática el cálculo de las características de la fisura y del proyectil.

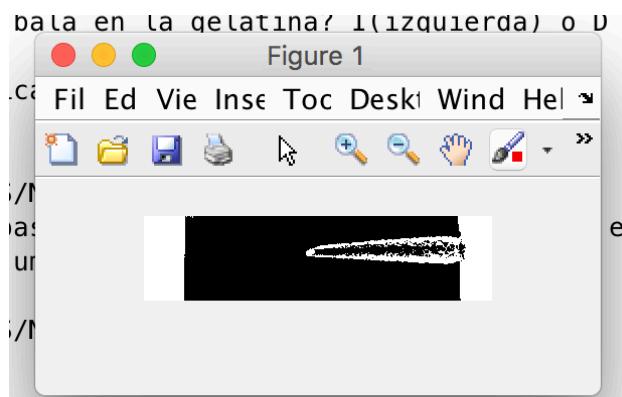


Figura 6-25: Video en el que se observa el avance del proyectil.

22. Una vez finalizado el programa aparece una tabla con el número de foto, el tiempo y el avance del proyectil tanto en pixeles como en mm.

Aplicaciones Técnicas de Visión Artificial a Balística de Efectos

Command Window				
informacion =				
Numero_foto	Tiempo	Longitud_fisura_1	Posicion_mm_1	
1	6.66666666666667e-06	0	0	
2	1.3333333333333e-05	0	0	
3	2e-05	4	5.60885608856089	
4	2.66666666666667e-05	5	7.01107011070111	
5	3.3333333333333e-05	7	9.81549815498155	
6	4e-05	9	12.619926199262	
7	4.66666666666667e-05	10	14.0221402214022	
8	5.3333333333333e-05	12	16.8265682656827	
9	6e-05	14	19.6309963099631	
10	6.66666666666667e-05	15	21.0332103321033	
11	7.3333333333333e-05	17	23.8376383763838	
12	8e-05	19	26.6420664206642	
13	8.66666666666667e-05	20	28.0442804428044	
14	9.3333333333333e-05	22	30.8487084870849	
15	0.0001	24	33.6531365313653	
16	0.000106666666666667	26	36.4575645756458	
17	0.000113333333333333	27	37.859778597786	
18	0.00012	29	40.6642066420664	
19	0.0001266666666666667	30	42.06642066420666	
20	0.000133333333333333	32	44.8708487084871	
21	0.00014	34	47.6752767527675	
22	0.0001466666666666667	35	49.0774907749078	
23	0.000153333333333333	37	51.8819188191882	
24	0.00016	39	54.6863468634686	
25	0.0001666666666666667	40	56.0885608856089	
26	0.000173333333333333	42	58.8929889298893	
27	0.00018	44	61.6974169741697	
28	0.0001866666666666667	45	63.09963099631	
29	0.000193333333333333	47	65.9040590405904	
30	0.0002	49	68.7084870848709	
31	0.0002066666666666667	50	70.1107011070111	
32	0.000213333333333333	52	72.9151291512915	
33	0.00022	53	74.3173431734317	
34	0.0002266666666666667	55	77.12171217712172	
35	0.000233333333333333	57	79.9261992619926	
36	0.00024	58	81.3284132841328	

Figura 6-26: Tabla con información sobre el proyectil.

23. Observar la tabla para detectar cualquier error.

Command Window				
194	0.0012933333333333	251	351.955719557196	
195	0.0013	252	353.357933579336	
196	0.00130666666666667	253	354.760147601476	
197	0.0013133333333333	254	356.162361623616	
198	0.00132	255	357.564575645756	
199	0.00132666666666667	256	358.966789667897	
200	0.0013333333333333	257	360.369003690037	
201	0.00134	258	361.771217712177	
202	0.00134666666666667	258	361.771217712177	
203	0.0013533333333333	259	363.173431734317	
204	0.00136	260	364.575645756458	
205	0.0013666666666667	261	365.977859778598	
206	0.0013733333333333	262	367.380073800738	
207	0.00138	263	368.782287822878	
208	0.00138666666666667	263	368.782287822878	
209	0.0013933333333333	264	370.184501845018	
210	0.0014	265	371.586715867159	
211	0.00140666666666667	268	375.793357933579	
212	0.0014133333333333	267	374.391143911439	
213	0.00142	268	375.793357933579	
214	0.00142666666666667	268	375.793357933579	
215	0.0014333333333333	270	378.59778597786	
216	0.00144	270	378.59778597786	
217	0.0014466666666667	311	436.088560885609	
218	0.0014533333333333	311	436.088560885609	
219	0.00146	311	436.088560885609	
220	0.0014666666666667	311	436.088560885609	
221	0.0014733333333333	311	436.088560885609	
222	0.00148	311	436.088560885609	
223	0.00148666666666667	311	436.088560885609	
224	0.0014933333333333	311	436.088560885609	
225	0.0015	311	436.088560885609	

Figura 6-27: Error en el cálculo de la fisura.

24. Entre la imagen 216 y 217 se observa un salto muy grande, esto quiere decir que el proyectil ya ha salido. Se ejecuta un parche para el cálculo de la gelatina al salir del bloque.

520	0.0034666666666667	9	12.619926199262
521	0.0034733333333333	9	12.619926199262
522	0.00348	9	12.619926199262
523	0.0034866666666667	9	12.619926199262

fx Observa algun error en la longitud de fisura? (S/N):S

Figura 6-28: Confirmación error.

25. Definir imagen a partir de la cual se produce el error.

521	0.0034733333333333	9	12.619926199262
522	0.00348	9	12.619926199262
523	0.0034866666666667	9	12.619926199262

Observa algun error en la longitud de fisura? (S/N):S
fx A partir de que imagen observa el error?:217

Figura 6-29: Definir imagen a partir de la cual se produce el error.

26. Ejecución del parche de forma automática.

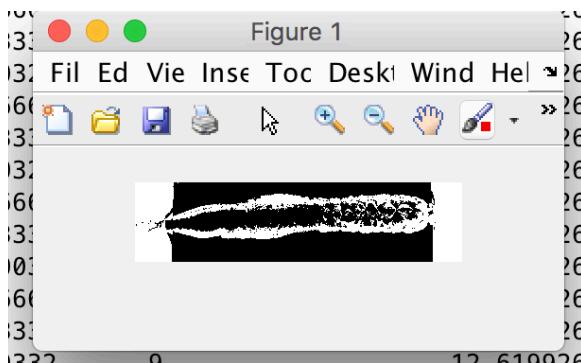


Figura 6-30: Video en el que se observa el avance del proyectil.

27. Aparece tabla con toda la información referente al proyectil y al orificio generado. Esta tabla es vertida a un Excel que queda guardado en el directorio en el cual se encuentran las imágenes.

Aplicaciones Técnicas de Visión Artificial a Balística de Efectos

o > Desktop > TFM-2017.ALVARO >

Command Window

```
informacion =
```

Numero_foto	Tiempo_s	Longitud_fisura_pixeles	Posicion_mm	Area_pixeles2
1	6.66666666666667e-06	0	0	0
2	1.3333333333333e-05	0	0	0
3	2e-05	4	5.60885608856089	9
4	2.66666666666667e-05	5	7.01107011070111	22
5	3.3333333333333e-05	7	9.81549815498155	37
6	4e-05	9	12.619926199262	49
7	4.66666666666667e-05	10	14.0221402214022	63
8	5.3333333333333e-05	12	16.8265682656827	78
9	6e-05	14	19.630963099631	93
10	6.66666666666667e-05	15	21.0332103321033	109
11	7.3333333333333e-05	17	23.8376383763838	126
12	8e-05	19	26.6420664206642	145
13	8.66666666666667e-05	20	28.0442804428044	161
14	9.3333333333333e-05	22	30.8487084870849	183
15	0.0001	24	33.6531365313653	201
16	0.000106666666666667	26	36.4575645756458	220
17	0.000113333333333333	27	37.859778597786	240
18	0.00012	29	40.6642066420664	262
19	0.000126666666666667	30	42.0664206642066	284
20	0.000133333333333333	32	44.8708487084871	302
21	0.00014	34	47.6752767527675	325
22	0.000146666666666667	35	49.0774907749078	346
23	0.000153333333333333	37	51.8819188191882	374
24	0.00016	39	54.6863468634686	398
25	0.000166666666666667	40	56.0885608856089	418
26	0.000173333333333333	42	58.8929889298893	443
27	0.00018	44	61.6974169741697	471
28	0.000186666666666667	45	63.09963099631	496
29	0.000193333333333333	47	65.9040590405904	520
30	0.0002	49	68.7084870848709	548
31	0.000206666666666667	50	70.1107011070111	573
32	0.000213333333333333	52	72.9151291512915	597
33	0.00022	53	74.3173431734317	620
34	0.000226666666666667	55	77.1217712177122	651

Figura 6-31: Información proyectil y orificio.

28. Se detalla la cantidad de energía intercambiada entre el proyectil y la gelatina.

522	0.00348	284	398.228782287823	128
523	0.0034866666666667	284	398.228782287823	132
La energía total cedida por el proyectil al cuerpo es:				
Energia_cedidatotal_J =				
7.127989829507956e+02				
fx >>				

Figura 6-32: Cantidad de energía intercambiada.

29. Se genera una gráfica con posición del proyectil frente al tiempo en base a las medidas obtenidas de las imágenes analizadas y se realiza una aproximación a polinomios de grado 1, 2, 3, 4.

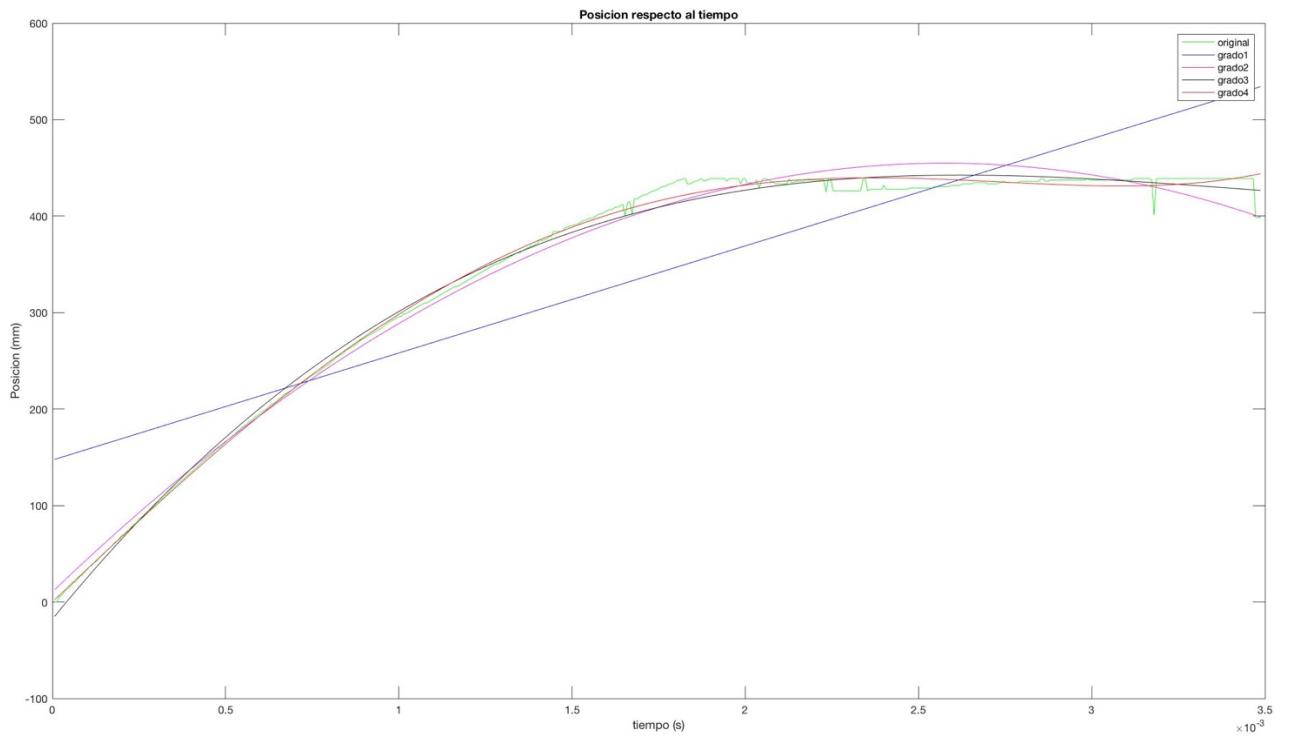


Figura 6-33: Gráfica posición frente al tiempo.

30. Gráficas de velocidad y aceleración frente al tiempo para el polinomio de grado 3 que es el que mejor se aproxima a la realidad.

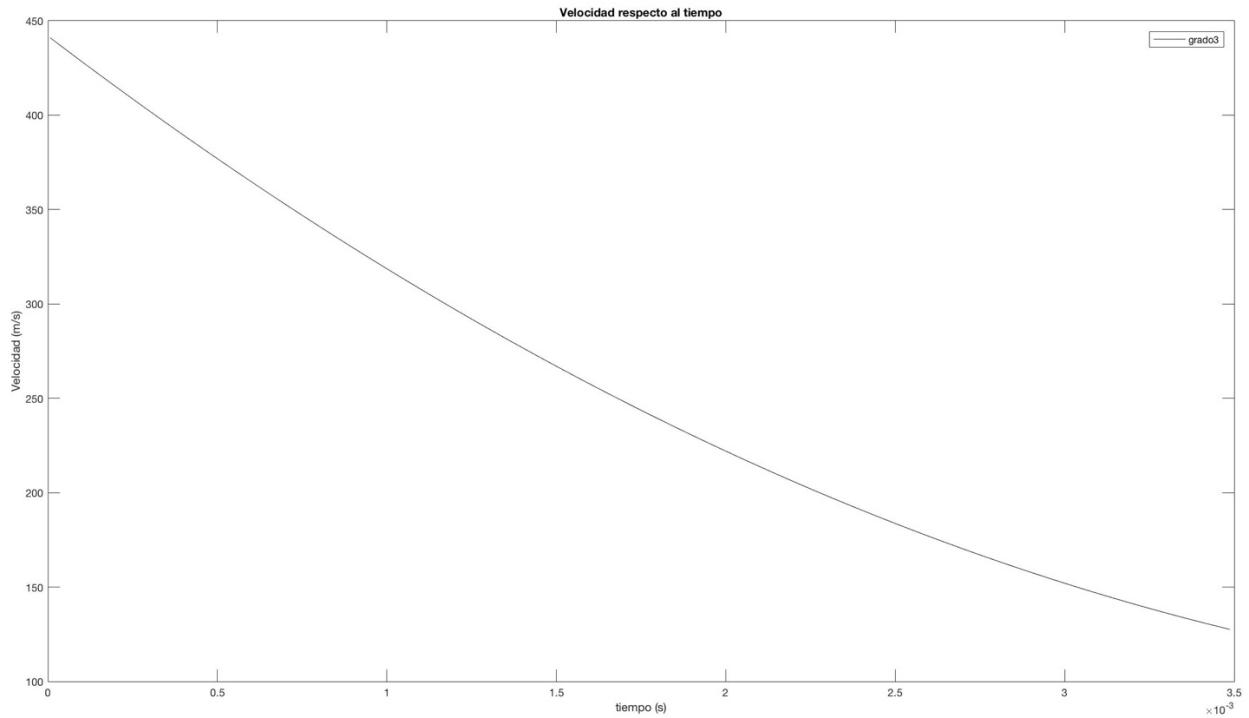


Figura 6-34: Gráfica de la velocidad frente al tiempo.

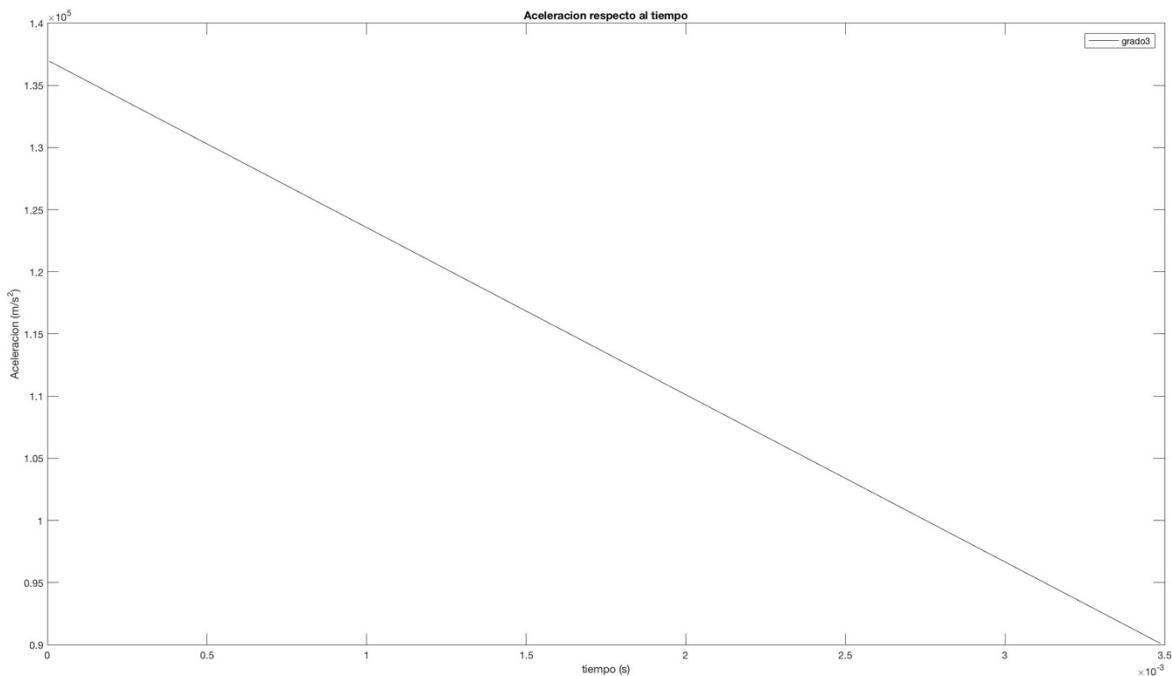


Figura 6-35: Gráfica de la aceleración frente al tiempo.

31. Realizando un análisis de las Figuras 6-33 a 6-35, se observa lo siguiente:

- a. El proyectil fmj, en base a la Figura 6-33, se observa que es capaz de atravesar completamente la gelatina, por lo que como se ha comentado anteriormente esto hace que el proyectil solo libera parte de la energía cinética en el cuerpo.
- b. En base a las Figuras 6-34 y 6-35 se observa un descenso con respecto del tiempo tanto de la velocidad como de la aceleración. Esto se debe al impacto del proyectil con los tejidos y su transferencia de energía.

7 PLANIFICACIÓN

La *Tabla 4* muestra las tareas que han sido realizadas, así como la duración y las fechas de comienzo y fin de cada una de ellas.

	Nombre de tarea	Duración	Comienzo	Fin	Predecesor
1	Creación de una aplicación para el estudio de la balística de efectos	83 días	mié 01/03/17	vie 23/06/17	
2	Definición del problema y propuesta del TFM	3 días	mié 01/03/17	vie 03/03/17	
3	Gestión del proyecto	3 días	mié 08/03/17	vie 10/03/17	2
4	Organización del proyecto	2 días	mié 08/03/17	jue 09/03/17	
5	Planificación	2 días	mié 08/03/17	jue 09/03/17	
6	Elaboración presupuesto	1 día	vie 10/03/17	vie 10/03/17	5
7	Especificación de requisitos	4 días	lun 13/03/17	jue 16/03/17	6
8	Análisis	15 días	vie 17/03/17	jue 06/04/17	7
9	Análisis Visión Artificial	5 días	vie 17/03/17	jue 23/03/17	
10	Análisis Balística Terminal	5 días	vie 24/03/17	jue 30/03/17	9
11	Análisis Matlab	5 días	vie 31/03/17	jue 06/04/17	10
12	Construcción	35 días	vie 07/04/17	jue 25/05/17	11
13	Testing	4 días	vie 26/05/17	mié 31/05/17	12
14	Realización de la documentación	20 días	lun 29/05/17	vie 23/06/17	

Tabla 4: Planificación de tareas.

A continuación, en la *Figura 7-1*, se observa el diagrama de Gantt obtenido a partir de la planificación de las tareas anteriores:

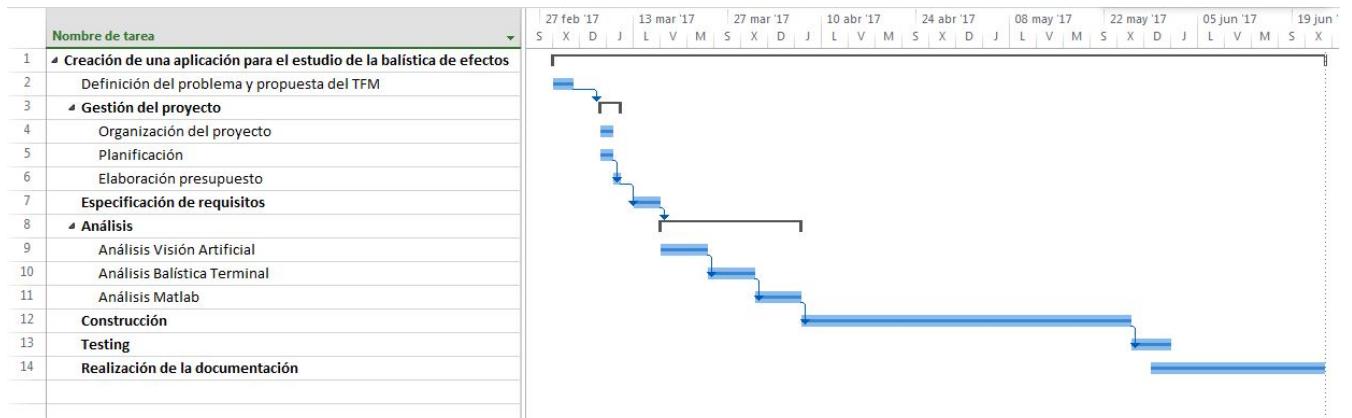


Figura 7-1: Diagrama de Gantt.

8 PRESUPUESTO

8.1 PARTIDA DE PERSONAL

En base a la *Tabla 4* se hace un desglose de los días que ambos ingenieros han dedicado al proyecto (*Tabla 5*).

<i>Estimación de las tareas por días</i>	<i>Ingeniero Junior</i>	<i>Ingeniero Senior</i>
<i>Definición del problema y propuesta del TFM</i>	2.5	0,5
<i>Gestión del proyecto</i>	2.5	0,5
<i>Especificación de requisitos</i>	3	1
<i>Análisis</i>	14	1
<i>Construcción</i>	33	2
<i>Testing</i>	2	2
<i>Realización de la documentación</i>	18	2
Total días	75	9

Tabla 5: Días dedicados por cada ingeniero a cada tarea.

Se ha tenido en cuenta que como media un ingeniero junior le dedica cada día 5 horas, mientras que un ingeniero senior le dedica 1 horas diaria.

En la *Tabla 6*, se detalla el coste por hora para cada uno de los ingenieros, así como el total de horas dedicadas y el coste total de dichas horas.

	<i>Coste/hora</i>	<i>Total horas en el proyecto</i>	<i>Coste Total (€)</i>
<i>Ingeniero Junior</i>	15	375	5.625
<i>Ingeniero Senior</i>	30	9	270
Total			5.895

Tabla 6: Coste de personal.

El coste de personal asciende a 5.895 €.

8.2 PARTIDA DE MATERIALES

En la *Tabla 7* se detallan los costes directos que son necesarios para poder realizar el proyecto junto con la cantidad y el valor de cada uno de ellos.

Para el cálculo del coste unitario de ambos ordenadores, Matlab y del Ms Project, se tiene en cuenta el coste total de estos y se divide entre la vida media (alrededor de 4 años).

<i>Material</i>	<i>Cantidad (meses)</i>	<i>Coste Unitario (€/mes*ud)</i>	<i>Coste Total(€)</i>
<i>Macbook Pro</i>	1	42	126
<i>PC Sobremesa Windows 10</i>	1	21	63
<i>Matlab</i>	2	42	252
<i>Microsoft Office 2016</i>	2	10.50	63
<i>Ms Project 2016</i>	2	25.30	151.8
<i>Photron FASTCAM Viewer (PFV) 3.6.7.0</i>	2	0	0
Total			655,80

Tabla 7: Costes Materiales.

El coste del material asciende a 655,80 €.

8.3 RESUMEN DE COSTES

En la *Tabla 8* se realiza un resumen de los costes totales del proyecto. Como coste añadido, se tiene en cuenta un beneficio del 8% y unos gastos generales del 15 % del valor total del proyecto.

PRESUPUESTO FINAL	<i>Importe (€)</i>
<i>Partida de Personal</i>	6.140,00
<i>Partida de Materiales</i>	655,80
Total partidas	6.795,80
<i>Gastos Generales (15%)</i>	1.019,37
<i>Beneficio (8%)</i>	543,67
Total (sin IVA)	8.358,83
<i>IVA (20%)</i>	1.671,77
TOTAL	10.030,60

Tabla 8: Coste total proyecto.

El coste total asciende a **10.030,60 €**.

9 CONCLUSIONES Y LÍNEAS FUTURAS

9.1 CONCLUSIONES

El objetivo principal de este trabajo, que era el de diseñar una aplicación que fuese capaz de adquirir información en tiempo real acerca del daño provocado por un proyectil en un cuerpo mediante el análisis de videos tomados, se ha conseguido con éxito.

La realización de este trabajo me ha permitido poner en práctica los conocimientos adquiridos durante mi etapa académica referentes al programa Matlab. Además, he conseguido ampliar los conocimientos referentes a este programa.

Este trabajo también me ha permitido familiarizarme con la balística de efectos, un campo para mí desconocido hasta ahora. Me ha dado la posibilidad de conocer de una manera más cercana el daño que un proyectil es capaz de generar al ser disparado contra un cuerpo y no solo por la velocidad de este sino también por la cavitación temporal generada.

Por último, mediante el desarrollo de este proyecto me he conseguido familiarizar con la visión artificial y me he dado cuenta de la gran cantidad de aplicaciones que pueden desarrollarse, permitiendo mejorar nuestro estilo de vida y hacerlo más seguro.

En conclusión, este proyecto ha sido una gran experiencia dentro del campo de la visión artificial y la balística de efectos.

9.2 LÍNEAS FUTURAS

En este epígrafe se definen una serie de ideas sobre posibles líneas a desarrollar en el futuro con el objetivo de continuar con el estudio realizado en el presente trabajo.

A continuación, se presentan diversas propuestas:

- En el programa para realizar los cálculos del volumen se ha supuesto que el orificio era un volumen de revolución, lo que no es correcto. Este error se podría subsanar mediante el uso de dos cámaras perpendiculares que estuviesen sincronizadas y grabasen el movimiento del proyectil simultáneamente.
- Con la ayuda de esta segunda cámara se podrían capturar los instantes en los que el proyectil está rotando o no.
- Añadir un algoritmo que permita calcular el cabeceo de la bala.

Aplicaciones Técnicas de Visión Artificial a Balística de Efectos

- Mejorar la iluminación durante la grabación de los videos de tal manera que no existan sombreados y se consigan videos de mejor calidad.
- Comparar los resultados obtenidos con la ayuda del programa desarrollado y los resultados obtenidos en el laboratorio de forma experimental.
- Se podría desarrollar una interfaz GUI una vez que el programa haya sido testeado por distintos usuarios que permita mejorar su aspecto visual.

10 BIBLIOGRAFÍA

10.1 LIBROS DE CONSULTA

- [1] De la Escalera Hueso, A. (2001). *Visión por Computador*. Madrid: Prentice Hall.

10.2 PÁGINAS WEB

- [2] <http://www.medigraphic.com/pdfs/cirgen/cg-2001/cg014i.pdf>
- [3] http://www.portalcultura.mde.es/Galerias/actividades/fichero/2015/Agosto/2015_Trab_PreDef_Mod_Investigacion.pdf
- [4] <http://cienciasforensesycriminalisticas.blogspot.com.es/>
- [5] <https://balistica-forense.wikispaces.com/file/view/BALISTICA+DE+EFFECTO.pdf>
- [6] <https://repositorio.unican.es/xmlui/handle/10902/8119>
- [7] <http://www.astrofisicayfisica.com/2012/06/que-es-el-espectro-electromagnetico.html>
- [8] http://oa.upm.es/14016/2/PFC_EDUARDO_LAORDEN_FITER_B.pdf
- [9] <http://juancarlosusomatl2015.weebly.com/definicion-matlab.html>
- [10] <https://es.mathworks.com/products/image.html>
- [11] http://asignatura.us.es/imagendigital/Matlab_PID_1314.pdf
- [12] <https://es.slideshare.net/lonely113/procesamiento-digital-de-imgenes-con-matlab>
- [13] https://es.mathworks.com/videos/computer-vision-with-matlab-for-object-detection-and-tracking-82036.html?elqsid=1487348051010&potential_use=Student
- [14] <https://es.mathworks.com/products/matlab.html>
- [15] <http://www.taringa.net/posts/info/18958106/Tipos-de-municion-FMJ-JHP-SWC-P-y-Mas.html>
- [16] <https://www.youtube.com/watch?v=HcVA2sGjgfo>
- [17] <https://es.mathworks.com/products/image/features.html - an%25C3%25A1lisis-de-im%25C3%25A1genes>

- [18] <https://es.mathworks.com/matlabcentral/answers/102711-how-do-i-break-out-of-nested-loops-using-the-break-command-in-matlab-7-7-r2008b>
- [19] <http://www.esi2.us.es/~jaar/Datos/FIA/T9.pdf>
- [20] <https://es.mathworks.com/help/matlab/ref/table.html>
- [21] https://es.mathworks.com/help/matlab/import_export/exporting-to-excel-spreadsheets.html - responsive_offcanvas
- [22] <https://www.youtube.com/watch?v=1SIxMh2ZR1Y>
- [23] <https://es.mathworks.com/help/symbolic/diff.html>
- [24] <https://www.stockarmas.com/test-balistico-rebotes-y-transferencia-de-energia/>

11 ANEXOS

11.1 ANEXO I: CÓDIGO

```

%% Programa que permite analizar el daño generado por un proyectil en un cuerpo.

format long
clear all; clc; close all;

%Esto sirve para guardar el directorio inicial y volver al finalizar el programa
diract = cd;

%Abrimos la carpeta que queramos en el cd, es decir, el directorio
%seleccionado si no elegimos ninguna devuelve 0
directorio0 = uigetdir(cd,'Elige la carpeta de imagenes que deseas cargar');

% si el directorio0 es igual 0 aparece el mensaje escrito a continuacion
if isequal(directorio0,0)
    msgbox('No ha elegido ninguna carpeta','MENSAJE','help')
    return
end

%Obtener imagenes de la carpeta que queramos utilizar
cd(directorio0)
lista = dir('*jpg');

%Obtenemos el numero de fotos que hay en la carpeta
[p,~] = size(lista);
if isequal(p,0)
    msgbox('No hay ninguna imagen en la carpeta especificada','MENSAJE','help')
    cd(diract)
    return
end

%Obtenemos el nombre de las fotos y los almacenamos en la matriz names
names = cell(p,1);
for i = 1:p
    names{i} = lista(i).name;
end

%Preguntamos el lado por donde entra la bala
lado = input('Cual es el lado por el que entra la bala en la gelatina? I(izquierda) o D(derecha):','s');

%Preguntamos la masa de la bala
masa = input('Masa de la bala (gr):');
%Pasamos la masa a kg
masa=masa/1000;

%Preguntamos el valor longitudinal de la gelatina balistica
longitud_gelatina_mm=input('Longitud real de la gelatina balistica (mm):');

%Frames por segundo
fps=input('Frames por segundo:');

%Cogemos la ultima imagen de la carpeta para cortarla y binarizarla
im_entrada = imread(names{end});

%Cortamos imagen y nos quedamos con lo que necesitamos
disp('Seleccione la fisura');
fis='S';
while fis=='S'
    %selecciono la parte de la probeta a analizar y la guardo en 'a' para hacer lo mismo en todas las fotos
    [I1,a]=imcrop(im_entrada);
    imshow(I1) %muestra la imagen cortada
    fis=input('Quiere volver a elegir la fisura? (S/N):','s');
end

%Calculo dimensiones imagen
[m,n] = size (I1);

%Creo una imagen de salida con fondo blanco con las dimensiones de la imagen de entrada
im_salida = ones(m,n);

%Determinamos el umbral y cortamos utilizando una imagen intermedia
lv='S';
while lv=='S'
    T=input('Determine el valor del umbral para pasar a binario (valores recomendados entre 40-60):');

```

Aplicaciones Técnicas de Visión Artificial a Balística de Efectos

```
close all;
%Con estos dos bucles voy recorriendo la imagen de entrada pixel por pixel y si el pixel de la
imagen de
%entrada es mayor que el umbral T lo convierto en negro en la imagen de salida y si es menor que T
en blanco

for v = 1:m
    for j = 1:n
        if(I1(v,j)) > T
            im_salida(v,j) = 0;
        else
            im_salida(v,j) = 1;
        end
    end
end

imshow(im_salida)
lv=input('Quiere volver a elegir el valor del umbral? (S/N):','s');

%Cortamos imagen y nos quedamos con lo que necesitamos
disp('Seleccione la fisura');
fis='S';
while fis=='S'
    %selecciono la parte de la probeta a analizar y la guardo en 'a' para hacer lo mismo en todas las
    fotos
    [I2,b]=imcrop(im_salida);
    imshow(I2) %muestra la imagen cortada
    fis=input('Quiere volver a elegir la fisura? (S/N):','s');
end

%%Calculamos la longitud de la gelatina con la primera imagen (Todavía no esta deformada)
im_entrada = imread(names{1});
I1=imcrop(im_entrada,a);

%Calculo dimensiones imagen
[m,n] = size (I1);

%Creo una imagen de salida con fondo blanco con las dimensiones de la imagen de entrada
im_salida = ones(m,n);

for v = 1:m
    for j = 1:n
        if(I1(v,j)) > T
            im_salida(v,j) = 0;
        else
            im_salida(v,j) = 1;
        end
    end
end

%Cortamos la imagen
I2=imcrop(im_salida,b);
%Calculo dimensiones imagen cortada
[y,x] = size (I2);

%Con este bucle obtengo el pixel negro mas a la derecha y mas arriba y lo
%guardo para despues medir la posicion respecto a el
flag=0; %hay que meter un flag porque el break solo detiene un for
for u = x:-1:1
    for w = 1:1:y
        if(I2(w,u)) == 0
            longitud_ref_derecha=u-1;
            altura_ref=w;
            flag=1;
            break
        end
    end
    if(flag==1)
        break
    end
end

%COn este bucle obtengo el pixel negro mas a la izquierda y mas arriba y lo
%guardo para despues medir la posicion respecto a el
flag=0; %hay que meter un flag porque el break solo detiene un for
for u = 1:1:x
    for w = 1:1:y
        if(I2(w,u)) == 0
            longitud_ref_izquierda=u+1;
            altura_ref_izquierda=w;
            flag=1;
            break
        end
    end
end
```

```

        end
    if(flag==1)
        break
    end
end

%Calculo de la longitud de la gelatina
longitud_gelatina = longitud_ref_derecha - longitud_ref_izquierda;
close all

%Declaramos las variables para crear la matriz
numero_foto=[];
tiempo=[];
longitud_fisura_1 = [];
longitud_fisura_mm_1 = [];
area=[];
area_mm2=[];
volumen=[];
volumen_mm3=[];
%Bala entra por el lado derecho de la gelatina
if lado == 'D' || lado == 'd'
    %Leemos las fotos y las tratamos
    for i = 1:p
        numero_foto=[numero_foto i];

        %Tiempo
        tiempo=[tiempo i/fps];

        %Leemos fotos
        A{i} = imread(names{i});
        I1=imcrop(A{i},a);

        %Calculo dimensiones imagen
        [m,n] = size (I1);

        %Creo una imagen de salida con fondo blanco con las dimensiones de la imagen de entrada
        im_salida = ones(m,n);

        %Con estos dos bucles voy recorriendo la imagen de entrada pixel por pixel y si el pixel de
        la imagen de
            %entrada es mayor que el umbral T lo convierto en negro en la imagen de salida y si es menor
            %que T en
            %blanco

        for v = 1:m
            for j = 1:n
                if(I1(v,j)) > T
                    im_salida(v,j) = 0;
                else
                    im_salida(v,j) = 1;
                end
            end
        end

        %Corto la imagen y me quedo con la parte que me interesa
        I2bis = imcrop(im_salida,b);

        %Vemos como avanza la fisura en forma de video
        imshow(I2bis)

        %Declaramos las variables para crear la matriz
        largo=[];
        alto=[];

        %Con este for consigo calcular la longitud de la fisura
        for u = longitud_ref_derecha:-1:1
            num_pixeles=0;
            if(I2bis(1,u)) == 0
                largo=[largo (longitud_ref_derecha-u)];
                for w = 1:1:y
                    if(I2bis(w,u)) == 0
                        num_pixeles=num_pixeles+1;
                    else
                        break;
                    end
                end
                for w = y:-1:1
                    if(I2bis(w,u)) == 0
                        num_pixeles=num_pixeles+1;
                    else
                        break;
                    end
                end
            alto=[alto (y-num_pixeles)];
        end
    end
end

```

Aplicaciones Técnicas de Visión Artificial a Balística de Efectos

```
end
if (num_pixeles >= y)
    largo=largo(1:(end-1));
    alto=alto(1:(end-1));
    break;
end
end

%Determinamos la longitud de la fisura
longitud_fisura_1 = [longitud_fisura_1 (longitud_ref_derecha-u-1)];
%Conversion de longitud fisura de pixeles a mm
longitud_fisura_mm_1 = (longitud_fisura_1 * longitud_gelatina_mm)/longitud_gelatina;

%Determinamos el area
area=[area sum(alto)];
%Conversion de area en pixeles2 a mm2
area_mm2 = (area * longitud_gelatina_mm)/longitud_gelatina;

%Determinamos el volumen
volumen=[volumen sum((pi*alto.^2)/4)];
%Conversion de volumen en pixeles3 a mm3
volumen_mm3 = (volumen * longitud_gelatina_mm)/longitud_gelatina;

end

else

%Bala entra por el lado izquierdo de la gelatina
%Leemos las fotos y las tratamos
for i = 1:p
    numero_foto=[numero_foto i];

    %Tiempo
    tiempo=[tiempo i/fps];

    %Leemos fotos
    A{i} = imread(names{i});
    I1=imcrop(A{i},a);

    %Calculo dimensiones imagen
    [m,n] = size (I1);

    %Creo una imagen de salida con fondo blanco con las dimensiones de la imagen de entrada
    im_salida = ones(m,n);

    %Con estos dos bucles voy recorriendo la imagen de entrada pixel por pixel y si el pixel de
    la imagen de
        %entrada es mayor que el umbral T lo convierto en negro en la imagen de salida y si es menor
    que T en
        %blanco

        for v = 1:m
            for j = 1:n
                if(I1(v,j)) > T
                    im_salida(v,j) = 0;
                else
                    im_salida(v,j) = 1;
                end
            end
        end
    end

    %Corto la imagen y me quedo con la parte que me interesa
    I2bis = imcrop(im_salida,b);

    %Vemos como avanza la fisura en forma de video
    imshow(I2bis)

    %Declaramos las variables para crear la matriz
    largo=[];
    alto=[];

    %Con este for consigo calcular la longitud de la fisura
    for u = longitud_ref_izquierda:1:x
        num_pixeles=0;
        if(I2bis(1,u)) == 0
            largo=[largo (u-longitud_ref_izquierda)];
            for w = 1:1:y
                if(I2bis(w,u)) == 0
                    num_pixeles=num_pixeles+1;
                else
                    break;
                end
            end
        for w = y:-1:1
```

Aplicaciones Técnicas de Visión Artificial a Balística de Efectos

```
        if(I2bis(w,u)) == 0
            num_pixeles=num_pixeles+1;
        else
            break;
        end
    alto=[alto (y-num_pixeles)];
end
if (num_pixeles >= y)
    largo=largo(1:(end-1));
    alto=alto(1:(end-1));
    break;
end
%Determinamos la longitud de la fisura
longitud_fisura_1 = [longitud_fisura_1 (u-1-longitud_ref_izquierda)];
%Conversion de longitud fisura de pixeles a mm
longitud_fisura_mm_1 = (longitud_fisura_1 * longitud_gelatina_mm)/longitud_gelatina;

%Determinamos el area
area=[area sum(alto)];
%Conversion de area en pixeles2 a mm2
area_mm2 = (area * longitud_gelatina_mm)/longitud_gelatina;

%Determinamos el volumen
volumen=[volumen sum((pi*alto.^2)/4)];
%Conversion de volumen en pixeles3 a mm3
volumen_mm3 = (volumen * longitud_gelatina_mm)/longitud_gelatina;

end
end

%Convertimos las matrices en columnas
Numero_foto = numero_foto';
Tiempo = tiempo';
Tiempo_s = Tiempo;
Longitud_fisura_1 = longitud_fisura_1';
Posicion_mm_1 = longitud_fisura_mm_1';
Area = area';
Area_mm2 = area_mm2';
Volumen = volumen';
Volumen_mm3 = volumen_mm3';
informacion = table(Numero_foto,Tiempo_s,Longitud_fisura_1,Posicion_mm_1)

%Declaramos las variables en caso de que usemos el parche
longitud_fisura_2 = [];
longitud_fisura_mm_2 = [];
close all

%Parche que permite resolver el error en el calculo de la longitud
fis=input('Observa algun error en la longitud de fisura? (S/N):','s');
inicio=p+1;
if fis=='S'
    %Seleccionamos a partir de que imagen editar
    inicio=input('A partir de que imagen observa el error?:');

    %Bala entra por el lado derecho de la gelatina
    if lado == 'D' || lado == 'd'
        for i = inicio:p
            %Leemos fotos
            A{i} = imread(names{i});
            I1=imcrop(A{i},a);

            %Calculo dimensiones imagen
            [m,n] = size (I1);

            %Creo una imagen de salida con fondo blanco con las dimensiones de la imagen de entrada
            im_salida = ones(m,n);

            %Con estos dos bucles voy recorriendo la imagen de entrada pixel por pixel y si el pixel de
            la imagen de
                %entrada es mayor que el umbral T lo convierto en negro en la imagen de salida y si es menor
                %que T en
                    %blanco

                    for v = 1:m
                        for j = 1:n
                            if(I1(v,j)) > T
                                im_salida(v,j) = 0;
                            else
                                im_salida(v,j) = 1;
                            end
                        end
                    end
                end
            end
        end
    end
end
```

Aplicaciones Técnicas de Visión Artificial a Balística de Efectos

```
end

%Corto la imagen y me quedo con la parte que me interesa
I2bis = imcrop(im_salida,b);

%Vemos como avanza la fisura en forma de video
imshow(I2bis)

%Declaramos las variables para crear la matriz
largo=[];
alto[];

%Con este for consigo calcular la longitud de la fisura
for w = 1:l:y
    num_pixeles=0;
    for u = 1:l:longitud_ref_derecha
        if(I2bis(w,u)) == 1
            num_pixeles=num_pixeles+1;
        else
            break;
        end
    end
    largo=[largo num_pixeles];
end
vmin=min(largo);
%Determinamos la longitud de la fisura
longitud_fisura_2 = [longitud_fisura_2 (longitud_ref_derecha-vmin)];
%Conversion de longitud fisura de pixeles a mm
longitud_fisura_mm_2 = (longitud_fisura_2 * longitud_gelatina_mm)/longitud_gelatina;
end

%Bala entra por el lado izquierdo de la gelatina
else

    for i = inicio:p
        %Leemos fotos
        A{i} = imread(names{i});
        I1=imcrop(A{i},a);

        %Calculo dimensiones imagen
        [m,n] = size (I1);

        %Creo una imagen de salida con fondo blanco con las dimensiones de la imagen de entrada
        im_salida = ones(m,n);

        %Con estos dos bucles voy recorriendo la imagen de entrada pixel por pixel y si el pixel de
        la imagen de
        %entrada es mayor que el umbral T lo convierto en negro en la imagen de salida y si es menor
        que T en
        %blanco

        for v = 1:m
            for j = 1:n
                if(I1(v,j)) > T
                    im_salida(v,j) = 0;
                else
                    im_salida(v,j) = 1;
                end
            end
        end

        %Corto la imagen y me quedo con la parte que me interesa
        I2bis = imcrop(im_salida,b);

        %Vemos como avanza la fisura en forma de video
        imshow(I2bis)

        %Declaramos las variables para crear la matriz
        largo=[];
        alto[];

        %Con este for consigo calcular la longitud de la fisura
        for w = 1:l:y
            num_pixeles=0;
            for u = x:-1:longitud_ref_izquierda
                if(I2bis(w,u)) == 1
                    num_pixeles=num_pixeles+1;
                else
                    break;
                end
            end
            largo=[largo num_pixeles];
        end
        vmin=min(largo);
```

Aplicaciones Técnicas de Visión Artificial a Balística de Efectos

```
%Determinamos la longitud de la fisura
longitud_fisura_2 = [longitud_fisura_2 (longitud_ref_derecha-vmin)];
%Conversion de longitud fisura de pixeles a mm
longitud_fisura_mm_2 = (longitud_fisura_2 * longitud_gelatina_mm)/longitud_gelatina;
end
end

%Convertimos las matrices en columnas
Longitud_fisura_2 = longitud_fisura_2';
Posicion_mm_2 = longitud_fisura_mm_2';

%Longitud con parche o sin parche
Longitud_fisura = [Longitud_fisura_1(1:(inicio-1))' Longitud_fisura_2'];
Posicion_mm = [Posicion_mm_1(1:(inicio-1))' Posicion_mm_2'];

%Dibujo grafica "Posicion respecto al tiempo"
tamano=get(0,'ScreenSize');
figure('Name','Posicion respecto al tiempo','NumberTitle','off','position',[tamano(1) tamano(2)
tamano(3) tamano(4)]);
plot(Tiempo,Posicion_mm,'g');
title('Posicion respecto al tiempo')
xlabel('tiempo (s)')
ylabel('Posicion (mm)')

%Con esto obtengo las ecuaciones polinomicas
disp('El polinomio aproximado a los puntos es:');
hold on

%Polinomio grado 1
p1 = polyfit(Tiempo,Posicion_mm,1);
yp1 = polyval(p1,Tiempo);
%Calculo correlacion polinomio grado 1
yav1=mean(Posicion_mm);
s1 = sum((Posicion_mm-yav1).^2);
s1bis = sum((Posicion_mm-yp1).^2);
z1 = 1 - s1bis/s1
%Dibujo grafica polinomio grado 1
plot(Tiempo,yp1,'b')

%Polinomio grado 2
p2 = polyfit(Tiempo,Posicion_mm,2);
yp2 = polyval(p2,Tiempo);
%Calculo correlacion polinomio grado 2
r2 = corrcoef(Posicion_mm,yp2);
yav2=mean(Posicion_mm);
s2 = sum((Posicion_mm-yav2).^2);
s2bis = sum((Posicion_mm-yp2).^2);
z2 = 1 - s2bis/s2
%Dibujo grafica polinomio grado 2
plot(Tiempo,yp2,'m')

%Polinomio grado 3
p3 = polyfit(Tiempo,Posicion_mm,3);
yp3 = polyval(p3,Tiempo);
%Calculo correlacion polinomio grado 3
r3 = corrcoef(Posicion_mm,yp3);
yav3=mean(Posicion_mm);
s3 = sum((Posicion_mm-yav3).^2);
s3bis = sum((Posicion_mm-yp3).^2);
z3 = 1 - s3bis/s3
%Dibujo grafica polinomio grado 3
plot(Tiempo,yp3,'k')

%Polinomio grado 4
p4 = polyfit(Tiempo,Posicion_mm,4);
yp4 = polyval(p4,Tiempo);
%Calculo correlacion polinomio grado 4
r4 = corrcoef(Posicion_mm,yp4);
yav4=mean(Posicion_mm);
s4 = sum((Posicion_mm-yav4).^2);
s4bis = sum((Posicion_mm-yp4).^2);
z4 = 1 - s4bis/s4
%Dibujo grafica polinomio grado 4
plot(Tiempo,yp4,'r')
legend('original','grado1','grado2','grado3','grado4')

%Calculo de la Velocidad
velocidad = diff(-p3);
Velocidad = polyval(velocidad, Tiempo);
Velocidad = Velocidad/1000;

%Calculo de la Aceleracion
aceleracion = diff(velocidad);
```

Aplicaciones Técnicas de Visión Artificial a Balística de Efectos

```
Aceleracion = polyval(aceleracion, Tiempo);
Aceleracion = Aceleracion/1000;

%Energia de la bala
Energia_proyectil=0.5*masa*Velocidad.^2;

%Dibujo grafica "Velocidad respecto al tiempo"
figure('Name','Velocidad respecto al tiempo','NumberTitle','off','position',[tamano(1) tamano(2)
tamano(3) tamano(4)]);
plot(Tiempo,Velocidad,'k');
title('Velocidad respecto al tiempo')
xlabel('tiempo (s)')
ylabel('Velocidad (m/s)')
legend('grado3')

%Dibujo grafica "Aceleracion respecto al tiempo"
figure('Name','Aceleracion respecto al tiempo','NumberTitle','off','position',[tamano(1) tamano(2)
tamano(3) tamano(4)]);
plot(Tiempo,Aceleracion,'k');
title('Aceleracion respecto al tiempo')
xlabel('tiempo (s)')
ylabel('Aceleracion (m/s^2)')
legend('grado3')

%Creamos la tabla donde almacenamos la informacion
Tiempo_s = Tiempo;
Longitud_fisura_pixeles = Longitud_fisura;
Area_pixeles2 = Area;
Volumen_pixeles3 = Volumen;
Velocidad_m_s = Velocidad;
Aceleracion_m_s2 = Aceleracion;
Energia_proyectil_J = Energia_proyectil;
Energia_cedida_J = -diff(Energia_proyectil);
Energia_cedida_J = [Energia_cedida_J;Energia_proyectil(end)];
informacion =
table(Numerofoto,Tiempo_s,Longitud_fisura_pixeles,Posicion_mm,Area_pixeles2,Area_mm2,Volumen_pixeles3,Volumen_mm3,Velocidad_m_s,Aceleracion_m_s2,Energia_proyectil_J,Energia_cedida_J)

%Energia cedida total cedida por el proyectil al cuerpo
disp('La energia total cedida por el proyectil al cuerpo es:');
Energia_cedidatotal_J=(Energia_proyectil_J(1)-Energia_proyectil_J(end))

%Exportamos a un excel
filename = 'informacionBala.xlsx';
writetable(informacion,filename,'Sheet',1,'Range','B2')

%Asi vuelve al directorio original
cd(diract)
```

11.2 ANEXO II: RESULTADOS PROYECTIL SP

En este anexo se adjuntan las gráficas referentes a la posición, velocidad y aceleración respecto al tiempo de un proyectil SP.

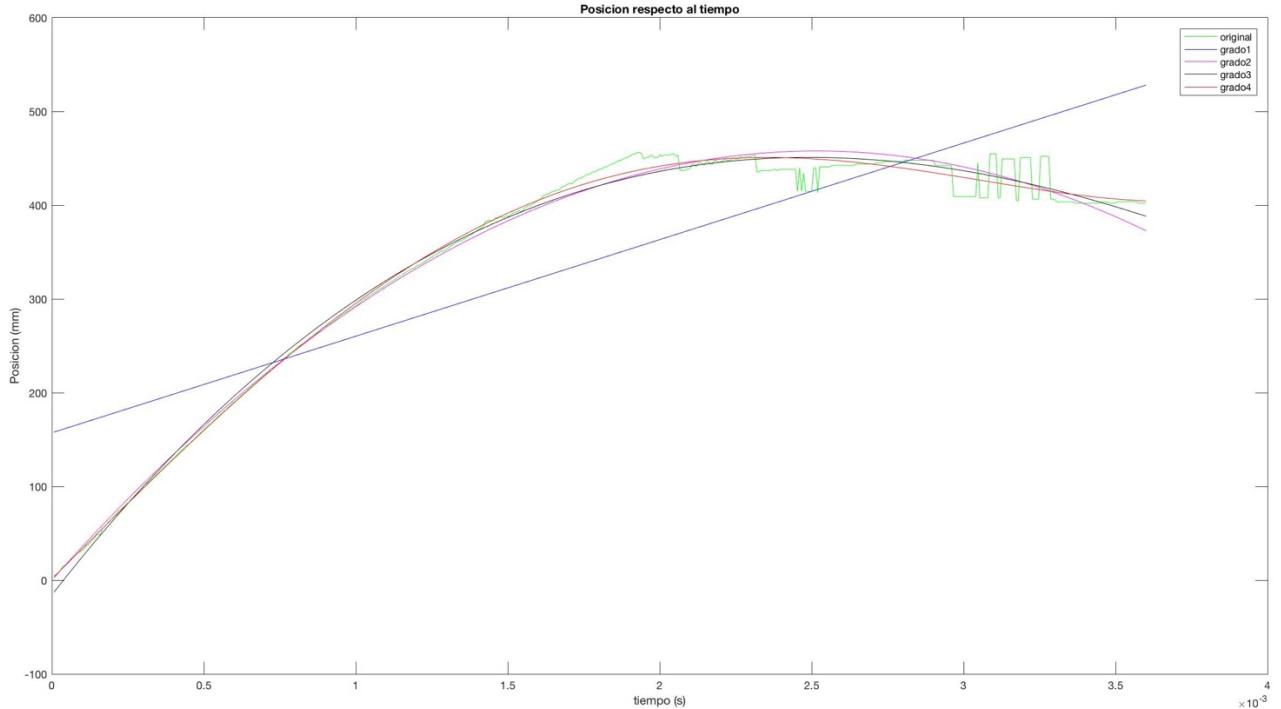


Figura II-1: Posición frente al tiempo.

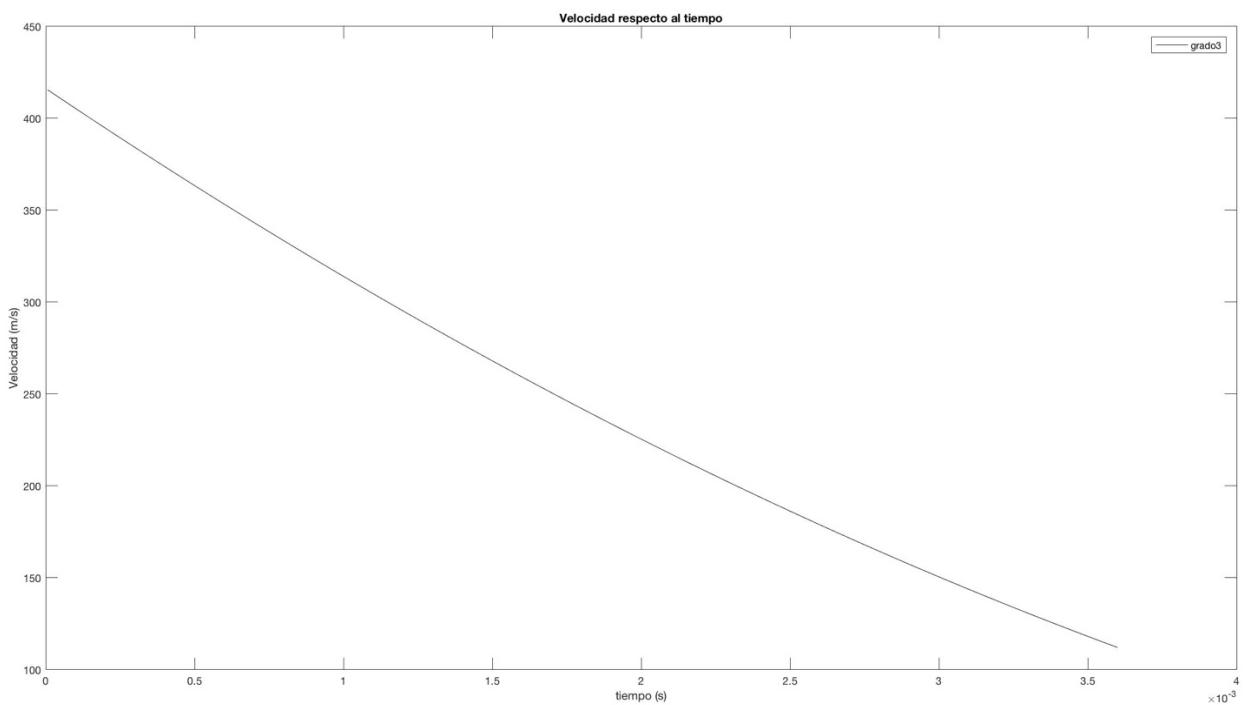


Figura II-2: Velocidad frente al tiempo.

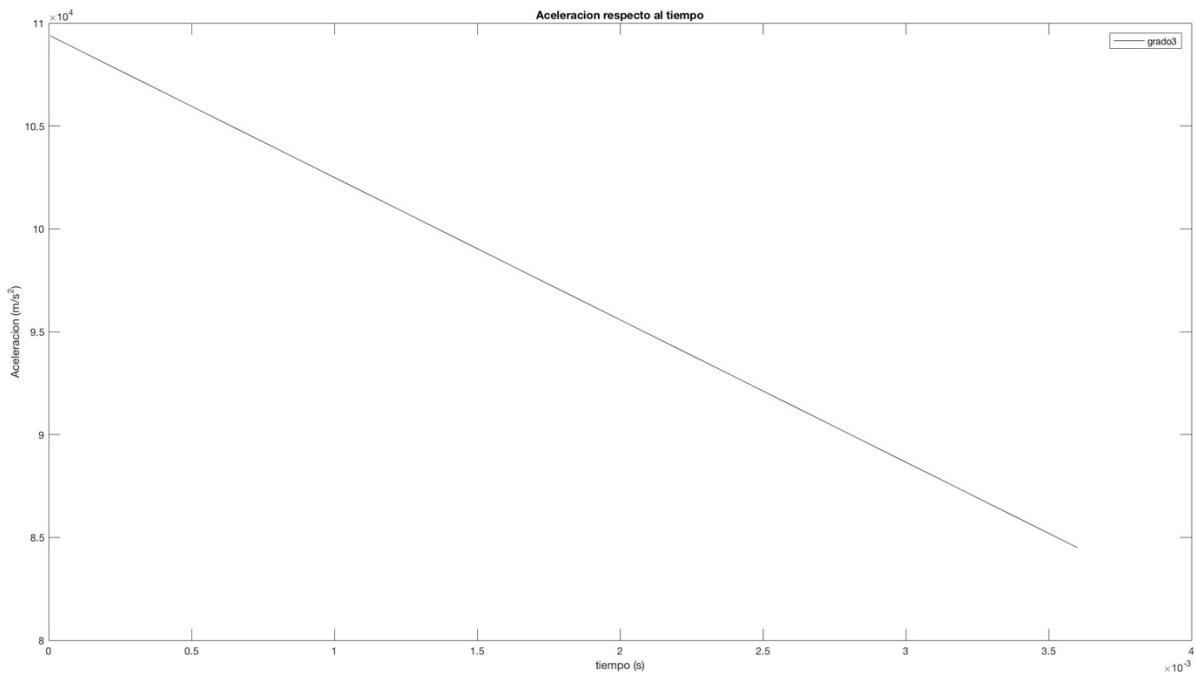


Figura 11-3: Aceleración frente al tiempo.

11.3 ANEXO III: RESULTADOS PROYECTIL HP

En este anexo se adjuntan las gráficas referentes a la posición, velocidad y aceleración respecto al tiempo de un proyectil HP.

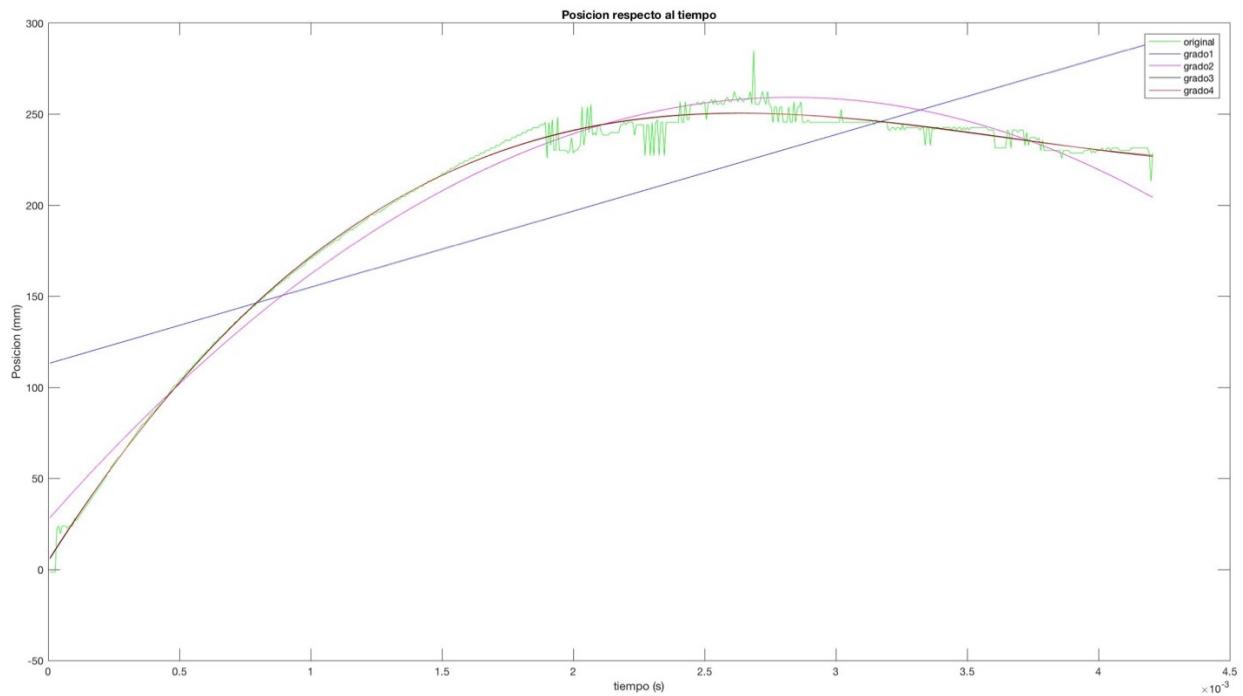


Figura 11-4: Posición frente al tiempo.

Aplicaciones Técnicas de Visión Artificial a Balística de Efectos

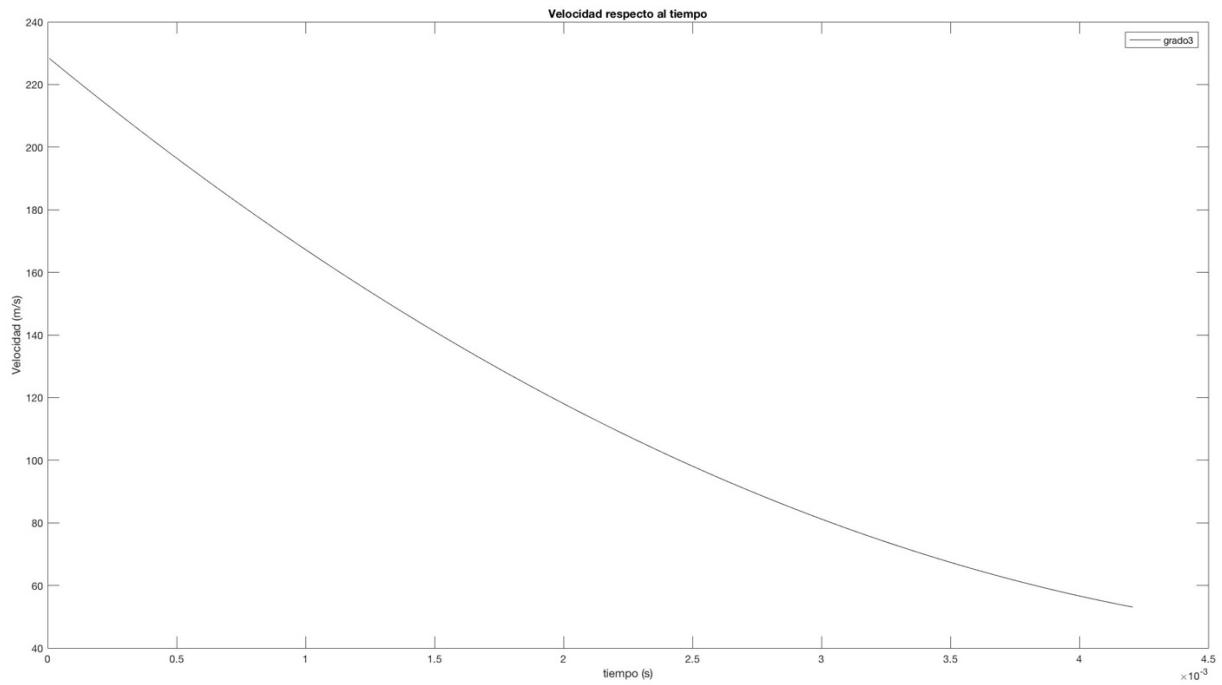


Figura 11-5: Velocidad frente al tiempo.

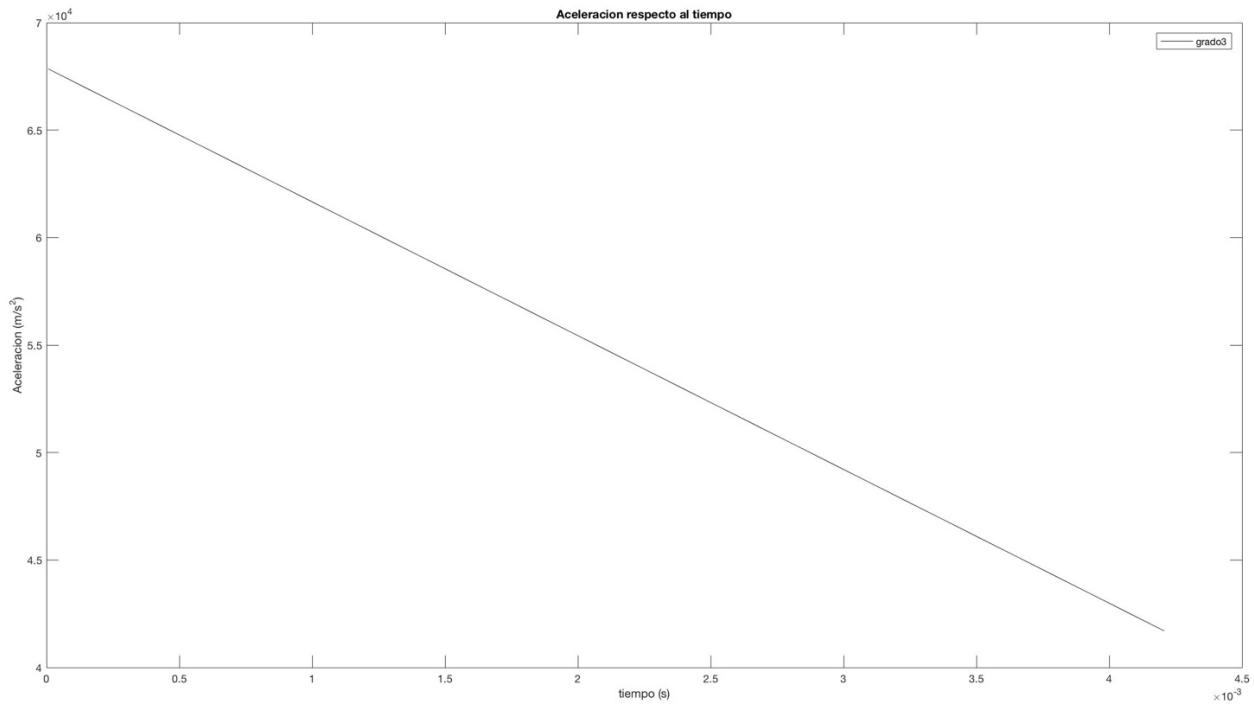


Figura 11-6: Aceleración frente al tiempo.