



Proyecto Fin de Carrera

Sistema de monitorización de posición para cámaras en rodajes 3D.

Autor/es

Javier Martínez Amo

Documento

Memoria

Director/es y/o ponente

Director: Jesús Ponce de León
Ponente: Antonio Bono Nuez

E.I.N.A

Convocatoria: Diciembre de 2011

Índice.

1. INTRODUCCION

1.1 Resumen

1.2 Requisitos de diseño

1.3 Estructura del documento

1.4 Introducción al 3D y la problemática del rodaje.

1.5 Proyectos anteriores de la empresa.

1.6 Aplicaciones

1.7 Antecedentes

1.7.1. Estereoscopía y cine 3D

1.7.2. Sensores de movimiento en todos los ejes

1.7.3. Placas de hardware libre: Arduino

1.8 Comentarios previos

1.8.1. Sobre las medidas absolutas y relativas

1.8.2. Sobre las limitaciones en los rodajes

1.8.3. Sobre el movimiento en el trípode

1.8.4. Sobre el uso de Arduino

1.9 Planificación

1.9.1. Etapas

1.9.2. Cronograma

2. ANALISIS DE SOLUCIONES

2.1 Decisiones de diseño

- 2.1.1. Tipos de movimiento y sensores
- 2.1.2. Magnetómetro y acelerómetro
- 2.1.3. Placa del sensor
- 2.1.4. Placa de microcontrolador
- 2.1.5. Alimentación
- 2.1.6. Arduino o Matlab

2.2 Descripción y documentación del hardware

- 2.2.1. Cómo funciona un acelerómetro
- 2.2.2. Cómo funciona un magnetómetro
- 2.2.3. La placa del sensor
- 2.2.4. La placa Arduino

2.3 Descripción y documentación del software

- 2.3.1. Introducción y esquema
- 2.3.2. Librería de lectura de datos mediante protocolo I2C entre Arduino y sensores.
- 2.3.3. Sketch de la placa Arduino
- 2.3.4. Librería para control de placas Arduino (arduino.m)
- 2.3.5. Funciones de calibración de sensores en Matlab.
- 2.3.6. Función de cálculo de inclinación
- 2.3.7. Función de cálculo de cabeceo
- 2.3.8. Representación gráfica en tiempo real.

3. CONCLUSIONES

- 3.1 Resultados finales
- 3.2 Futuras mejoras

3.3 Conclusión personal

4. REFERENCIAS Y GLOSARIO

4.1 Referencias

4.2 Software y gestión

4.3 Glosario

 <p>Escuela de Ingeniería y Arquitectura Universidad Zaragoza</p>	<p>Sistema de monitorización de la posición de cámaras en rodajes 3D</p>	<p>Memoria 14/11/2011</p>
--	--	-------------------------------

1. INTRODUCCIÓN

Nota previa: en adelante los superíndices ⁿ con numeración árabe indican referencias en el glosario, sección 4.3. Los números encerrados entre corchetes [n] se refieren a bibliografía en la sección de referencias 4.1.

1.1 Resumen.

El presente proyecto ha sido llevado a cabo para la empresa productora de cine 3+D Entertainment S.L. como parte de su programa de investigación y desarrollo.

Se pretende desarrollar un sistema de monitorización¹ del movimiento del rig² que sustenta la o las cámaras de video usadas en un rodaje profesional.

Concretamente se debe lograr averiguar la posición absoluta en tiempo real de la cámara en todos los ejes posibles con la precisión adecuada.

Los objetivos centrales se podrían resumir del siguiente modo, haciendo una distinción entre hardware y software:

a) Obtener las medidas de posición a partir de un magnetómetro y un acelerómetro así como su adecuado tratamiento digital y calibración de los dispositivos.

b) Desarrollar una interface con Matlab compatible con Arduino y que permita la integración de otros proyectos y mejoras posteriores.

1.2 Requisitos de diseño.

-Prototipo de sistema electrónico con sensor/es y placa de

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

microcontrolador Arduino para monitorizar movimientos de cámaras encima de un trípode desde un ordenador.

-Es imprescindible que el hardware sea ampliable posteriormente con nuevos sensores. No realizar un sistema cerrado.

-Software de monitorización y control también adaptable a posibles añadidos posteriores (más sensores, control de motores, etc) preferiblemente en Matlab.

-El software debe poder operar en los principales sistemas operativos. Especialmente en Windows y Mac OS.

-Tamaño máximo de la PCB del sensor del orden del tamaño de la placa Arduino.

-Ángulo de inclinación máximo del sensor/es en torno a +- 45º en torno al eje.

-Capacidad de realización de unas 25 mediciones del sensor por segundo, incluyendo tratamiento de la señal.

-Error máximo de 2º en cualquier medida angular en términos absolutos.

-Error máximo del 1% en las medidas en términos relativos.*

**Aclaraciones en apartado 1.7.1*

CARACTERÍSTICAS FÍSICAS:

-Para el montaje del prototipo no será necesario el uso de ningún encapsulado, puesto que dificultaría eventuales interacciones posteriores. No obstante sería conveniente dotar al prototipo de cierta

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

estructura que le permita mantenerse estable.

CARACTERÍSTICAS AMBIENTALES DE FUNCIONAMIENTO PREVISTAS:

- Rango de temperaturas entre 0°C y 75°C.
- Rango de humedad relativa entre 5% y 90%.

1.3 Estructura del documento.

El presente documento ha sido estructurado de la siguiente manera:

En primer lugar una amplia introducción en la que se trata la problemática a la que se pretende dar solución con este proyecto y con la que se pretenden poner en contexto las decisiones tomadas. También en este apartado se encuentran diversas aclaraciones y conceptos que es recomendable conocer antes de pasar al desarrollo técnico.

A continuación se pasa a describir las decisiones de diseño tomadas y sus causas. Seguidamente comienza una descripción desde el punto de vista técnico de todos los aspectos del proyecto. En esta sección se encuentran todos los detalles concretos referentes al hardware y en especial el software del proyecto.

En la tercera sección se muestran los resultados y conclusiones pertinentes, así como una relación de las posibles ampliaciones que

 <p>Escuela de Ingeniería y Arquitectura Universidad Zaragoza</p>	<p>Sistema de monitorización de la posición de cámaras en rodajes 3D</p>	<p>Memoria 14/11/2011</p>
--	--	-------------------------------

puede recibir el sistema en el futuro.

En último lugar se encuentran las referencias y bibliografía, un glosario con definiciones de posibles palabras de interés y por último una sección de anexos.

1.4 Introducción al 3D y la problemática del rodaje.[1]

El rodaje estereoscópico requiere de dos cámaras exactamente iguales situadas a una cierta distancia interocular (típicamente en torno a la separación de los ojos humanos) y con un cierto ángulo de convergencia. Algunas definiciones previas:

-Se define como *plano de convergencia* el plano donde converge la mirada de ambos ojos/cámaras. Por tanto es la única zona en la que la imagen sería la misma para ambos.

-El *paralaje* se define como la desviación angular de la posición aparente de un objeto, dependiendo del punto de vista elegido. En estereoscopía hace referencia a la distancia espacial de representación de un mismo objeto para el ojo/cámara izquierdo y derecho en un mismo instante temporal.

Todos los objetos que están por detrás del plano de convergencia tienen por convenio un paralaje negativo. Todo lo que está por delante tiene paralaje de signo contrario. Por último, todos los objetos situados en el plano de convergencia tienen un paralaje nulo.

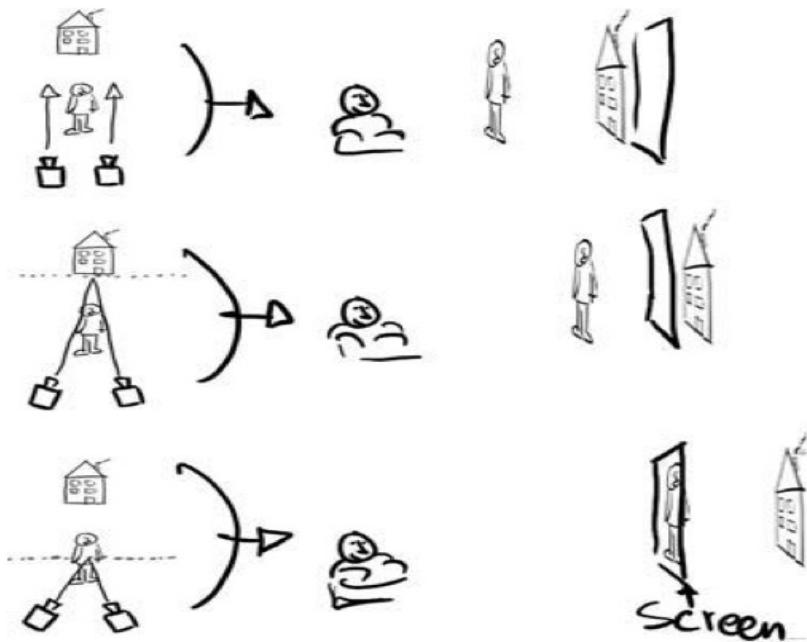
 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

El *ángulo de convergencia* por su parte es la desviación de las cámaras respecto a la disposición de paralelismo. Dicho de otro modo, es el ángulo formado por las cámaras y el plano de convergencia.

Los parámetros de 3D modificables en el rodaje serían pues la distancia interocular y el ángulo de convergencia. Conviene ilustrar gráficamente las consecuencias de cada uno de estos parámetros:

En primer lugar el ángulo de convergencia influye en la posición de los objetos respecto de la pantalla. Ésta y el plano de convergencia siempre van a coincidir, por tanto el lugar donde se sitúe el eje será el que aparezca en la pantalla durante la reproducción.

Cuando el ángulo de convergencia es cero, el plano de convergencia se sitúa en el horizonte. En consecuencia todos los objetos quedan por delante de del eje y por tanto de la pantalla.



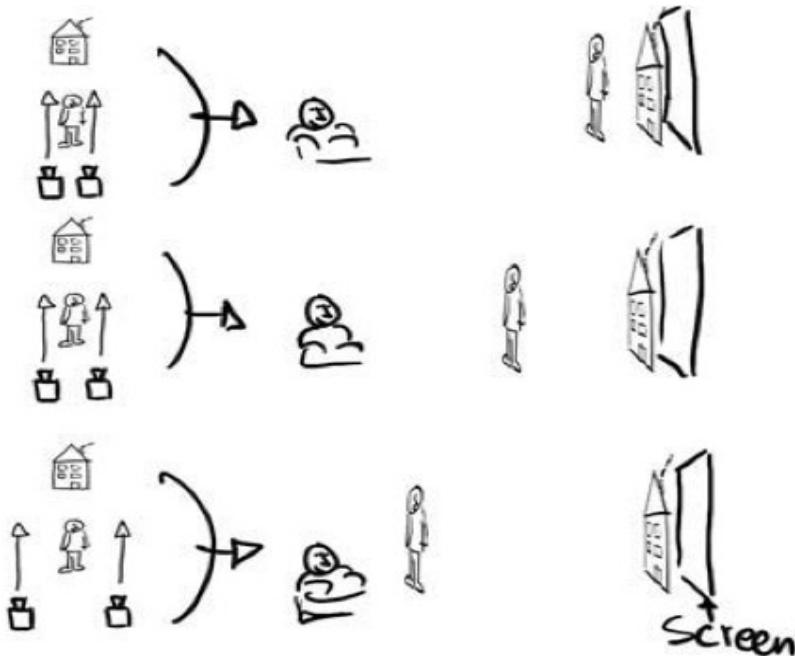
Conforme el ángulo de convergencia se vuelve más obtuso, el plano de convergencia avanza hacia el observador. En consecuencia se pueden situar objetos delante de la pantalla, pero el horizonte y objetos lejanos siempre aparecerán detrás.

El último caso expuesto consiste en cerrar el ángulo hasta situar el plano de convergencia en el objeto más adelantado del espacio de rodaje. Automáticamente el objeto quedaría en la pantalla, mientras que todo el resto de la escena transcurriría detrás de la misma.

En cualquier caso también se podría situar todos los objetos detrás de la pantalla con esta misma configuración.



El segundo parámetro configurable es la distancia interfocal. Esta no afectará a la posición de los objetos respecto de la pantalla, sino a la distancia de dichos objetos entre sí.



Una mayor distancia interfocal también contribuye a mejorar la percepción de objetos curvos, ya que pueden captar una mayor superficie del objeto entre las dos cámaras.

Como se ha podido observar, basta con estos parámetros para alcanzar todas las posibles combinaciones de posicionamiento de objetos en el plano. Sin embargo pueden aparecer una serie de problemas que se expresan a continuación.

El primero es que los movimientos de cámaras y de objetos o personajes en cada escena hacen imposible mantener un buen ajuste

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

del plano de convergencia, ya que es complicado modificarlo en tiempo real. Se necesita conocer cómo varía la posición de la cámara para poder realizar un ajuste fino y automático del ángulo de convergencia y la separación entre lentes.

Expresado con un ejemplo:

Las cámaras apuntan a un personaje, que está a cierta distancia. Deseamos el plano de convergencia justo donde está ese personaje, por lo tanto tenemos unos parámetros de configuración determinados.

Ahora el personaje va hacia la izquierda y se aleja. Podríamos seguir el movimiento con la cámara, y asignar a cada posición de la misma nuevos parámetros. De este modo al repetir la escena no habría que realizar los ajustes manualmente.

Esto es sólo un ejemplo de una de las necesidades que pueden ser solventadas con el presente proyecto. No obstante es útil o imprescindible en otras situaciones. Véase el apartado 1.5.

1.5 Proyectos anteriores de la empresa.

La empresa contratante, *3+D Entertainment*, es una productora de cine que, entre otros propósitos, tiene como objetivo desarrollar un completo sistema de rodaje 3D profesional de bajo coste.

El primero de los proyectos se realizó con ARSTIC, spinoff de la Universidad de Zaragoza. En él se consiguió un sistema de visión por

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

computador que capturaba imagen a través de las propias cámaras del rig 3D. Se situaba una fiducial⁵ en el lugar dónde se quería poner el plano de convergencia y mediante una serie de algoritmos el computador averiguaba a qué distancia estaban las cámaras de dicho eje y la separación entre ellas, así como el ángulo de convergencia que formaban.

Se completó el proyecto con una segunda parte que incluía dos motores paso a paso gestionados desde el mismo computador. Con los datos llegados a través del sistema de visión por computador, las cámaras se desplazaban lo necesario para situar el plano de convergencia donde se encuentra la fiducial. De este modo le lograba una primera posición desde la que comenzar la escena con los parámetros ajustados de forma automática.

Al disponerse de motores, se implementó también un software para gestionar su movimiento. Dichos motores se manejan desde una placa Arduino que a su vez se comunica con un PC. Además se incluye la posibilidad de memorizar movimientos, de tal manera que se automaticen los desplazamientos.

Hasta aquí los proyectos desarrollados de antemano. En este momento se necesita que los movimientos de las cámaras sean conocidos para que el control de motores se pueda hacer de manera automática y no manualmente, entre otras cosas que se comentarán más adelante.

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

Por otro lado se planteó una solución cerrada para el software porque se tuvo que aplicar durante un largometraje, por lo que era conveniente un programa independiente y funcional que se pudiera instalar en distintos ordenadores portátiles. Sin embargo no es una decisión muy adecuada para el terreno de la investigación, ya que todo el tiempo se implementan funcionalidades nuevas, en un proceso constante de mejora, experimentación y pruebas. Debido a esto se ha tenido que plantear la unificación en Matlab de la gestión de las placas Arduino, para el tiempo que dure el desarrollo e investigación.

1.6 Aplicaciones.

La primera aplicación que se busca con estos datos es el control en tiempo real de motores paso a paso utilizados para manejar la separación entre cámaras y ángulo de convergencia (proyecto anterior de la empresa). Como se ha explicado antes ya se tiene fijada una posición inicial con su respectivo plano de convergencia y separación interfocal. Los sensores nos permitirían entonces conocer las variaciones de posición y tomar decisiones sobre los motores, típicamente mediante un software de control.

Por otro lado, las aplicaciones en cine 3D son varias. De hecho ya se están usando sistemas parecidos en producciones de Hollywood. La mayor diferencia reside en que la propuesta de este proyecto es mucho más económica, tratando de mantener la funcionalidad.

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

Aquí otras cuestiones para las que es importante disponer de sensores:

Ayuda a la posproducción⁴ y efectos especiales.

En el caso del 3D se complica en gran medida puesto que se hace necesario aplicar todos los efectos a las grabaciones de ambas cámaras, que están ligeramente separadas y por tanto la imagen no es exactamente igual. Saber cuando las cámaras han cometido algún error y por qué para solucionarlo en posproducción es por tanto de suma importancia.

Posibilidad de planificación y robotización de planos⁴ (repetibilidad).

La mayoría de los planos en los rodajes requieren muchas repeticiones. Hacerlo manualmente en 2D no supone mayor problema. Sin embargo en 3D es imprescindible realizar un ajuste más fino de los parámetros de las cámaras, especialmente si estos van a variar a lo largo del plano. Conocer los movimientos de las cámaras es el primer paso hacia la robotización de los movimientos, que permitiría repetir con exactitud, toma tras toma, en los momentos más delicados del rodaje.

-Inclusión de escenografía renderizada⁶ en los monitores de campo⁷.

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

En las películas con gran cantidad de escenas con croma³ es imperativo que el director pueda tener una idea de lo que se está rodando a través de los monitores de campo. Son necesarios sensores para poder variar la simulación de los escenarios según se produzcan o no desplazamientos de las cámaras.

Es recomendable leer las referencias del glosario para información más detallada.

1.7 Antecedentes

1.7.1 La estereoscopía⁸ y el cine 3D [2]

El 3D es una tecnología relativamente nueva como fenómeno filmográfico mainstream⁹. Sin embargo, la estereoscopía es utilizada en el cine desde los años 20 del pasado siglo. Su uso se vio limitado por las capacidades tecnológicas de la época, que no han sido subsanadas hasta auge del cine tridimensional de la primera década del siglo XXI.

Los desafíos tecnológicos han acompañado al cine tridimensional desde su cuna. Incluso en la primera explosión del cine estereoscópico en los años 40, fue la complejidad técnica lo que disparó a la baja su rentabilidad y lo condenó al olvido. No obstante, la mayor lacra de este tipo de cine reside en el rodaje. Se hace necesaria una tecnología muy avanzada para lograr la fluidez y posibilidades artísticas del cine tradicional. En esta misma línea se están buscando soluciones para la emisión en directo de contenidos 3D para impulsar su empleo en

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

televisión.

El principal referente técnico que se ha tenido en cuenta para este y todos los proyectos de la empresa es la solución adoptada por James Cameron durante el rodaje de "Avatar". Consiste en una monitorización de todos los movimientos de la cámara, y la utilización de estos datos para robotización de cámaras (incluso en tiempo real), repetición y aplicación correcta de los efectos especiales y la posproducción. Por supuesto la solución adoptada por Cameron fue una costosísima obra de ingeniería que pocos estudios europeos se pueden permitir adquirir o desarrollar.

En Europa la industria del 3D todavía no ha alcanzado una mínima cuota de mercado y la tecnología apenas está desarrollada. Bajo esta premisa se hace muy necesario desarrollar sistemas más sencillos y económicos para el contexto de la televisión o el cine de mediano presupuesto.

1.7.2 Sensores de movimiento en todos los ejes.

La medida de los movimientos en cualquiera de los ejes cartesianos y polares no es algo nuevo, de hecho es algo muy extendido en el terreno de la aeronáutica. Es precisamente en los aviones donde se normalizó la nomenclatura de los ejes XYZ y el nombre de los giros (tilt, roll y heading¹⁰ por sus nombres anglosajones). Desafortunadamente, aunque el principio físico es el mismo, los sensores aeronaúticos son muy caros y voluminosos por lo

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

que no tienen aplicación posible en el caso que nos ocupa.

A una escala más similar a la nuestra nos encontramos con soluciones electrónicas para teléfonos móviles y consolas. Especial interés merecen sensores de movimiento como el WiiMote¹¹ (basado en acelerómetros, giroscopios e infrarrojos) o el Kinect¹² (basado en visión por computador e infrarrojos). Precisamente de este tipo de dispositivos fueron los que inspiraron en un principio este proyecto. En cualquier caso la aplicación es bien distinta y ninguno de los disponibles cumplía con las especificaciones necesarias para resolver los problemas planteados, como se explica en el apartado 2.1.1.

1.7.3 Placas de hardware libre¹³: Arduino.

Arduino es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y uso sencillo. En origen fue creado para artistas, diseñadores, aficionados y en general personas interesadas en crear entornos u objetos interactivos.

Arduino puede tomar información del entorno a través de sus pines de entrada de toda una gama de sensores y puede controlar aquello que le rodea como luces, motores y otros actuadores. El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación arduino (basado en Wiring¹⁴) y el entorno de desarrollo Arduino (basado en Processing¹⁵). Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de conectar a un

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

ordenador, si bien tienen la posibilidad de hacerlo y comunicar con diferentes tipos de software.

Fuente www.arduino.cc

Esta plataforma de desarrollo, que empezó siendo un proyecto para aficionados, se está convirtiendo en un referente a nivel mundial. Esto ha ocurrido gracias a una curva de aprendizaje muy cómoda, acompañada de grandes posibilidades de expansión.

Es difícil encontrar un límite para esta plataforma puesto que la comunidad no deja de aportar nuevos programas y *shields*. Estos últimos son placas PCB adaptadas para colocar directamente sobre Arduino y dotarle de infinidad de posibilidades.

Como ejemplo cabe destacar el caso de Telefónica I+D. Esta empresa ha desarrollado su propio *shield* para añadir la funcionalidad GPRS-GSM a Arduino, consiguiendo comunicarse con teléfonos móviles u otras placas similares.[16]

1.8 Comentarios previos

1.8.1 Sobre las medidas absolutas y relativas:

En los rodajes ya se dispone de un sistema que facilita información sobre la situación inicial del rig (y por tanto del sensor). Sin embargo no se puede conocer la posición final, ni tampoco posibles posiciones intermedias en las que la cámara se detenga o cambie el ritmo de su

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

desplazamiento.

Esto justifica que las medidas sean absolutas y no una serie de diferencias entre medidas de posición (velocidad). El uso de sensores de velocidad, como los giroscopios, acumulan error conforme se realizan movimientos salvo que conozcamos las posiciones inicial y final de manera absoluta.

No tiene demasiada importancia si de la medida que se desprende que la inclinación es cero, cuando en realidad existe una pequeña inclinación, y lo mismo ocurre con la posición de la cámara respecto al norte (cabeceo). Lo que es fundamental es que esos puntos sean siempre los mismos, así como la distancia entre unos y otros.

Ejemplo: Si la inclinación de la cámara cuando marca (0,0,0) es en realidad x:0.5°, y:0.2°, z:0.1°, es indispensable que el dato sea fijo siempre. De haber un error, este debe ser constante. La importancia de esto reside en la repetibilidad de los movimientos. Debe poderse regresar al punto inicial, sea cual sea.

1.8.2 Sobre las limitaciones en los rodajes:

Hay una serie de cuestiones que deben tenerse en cuenta en los rodajes y por tanto han delimitado bastante el camino del desarrollo del proyecto.

En primer lugar la intemperie es una cuestión a tener en cuenta. Por ejemplo los sensores de infrarrojos en exteriores no pueden

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

usarse, por lo que fueron descartados automáticamente. Sin embargo un sensor basado en campos magnéticos sólo se ve afectado de forma crítica con imanes, líneas de alta tensión muy cercanas y tormentas. Si se tienen en cuenta estas precauciones se puede trabajar con él.

También es necesario mirar por la humedad y la temperatura. Los requisitos no tienen por qué ser muy extremos, pero en un momento determinado se podría llegar a los límites soportables por los componentes electrónicos. Esto se ha de tener en cuenta tanto durante los rodajes como en el caso de encapsulados del producto final (no del prototipo). Para malas condiciones climáticas hay que asegurar el aislamiento y ventilación del sensor, por ser la parte más delicada.

Por último se hace necesario contar con la posible ausencia o escasez de conexiones a la red eléctrica durante ciertos momentos del rodaje. Para evitar problemas lo mejor es barajar estas dos posibilidades:

-Alimentar desde un ordenador portátil. Siendo el uso de estos dispositivos imprescindible, siempre se dispondrá de alimentación.

-Diseñar un alimentador basado en pilas de clase AA o AAA. Todo equipo de producción toma siempre la precaución de adquirir este tipo de fungibles previamente a los rodajes, tanto interiores como exteriores.

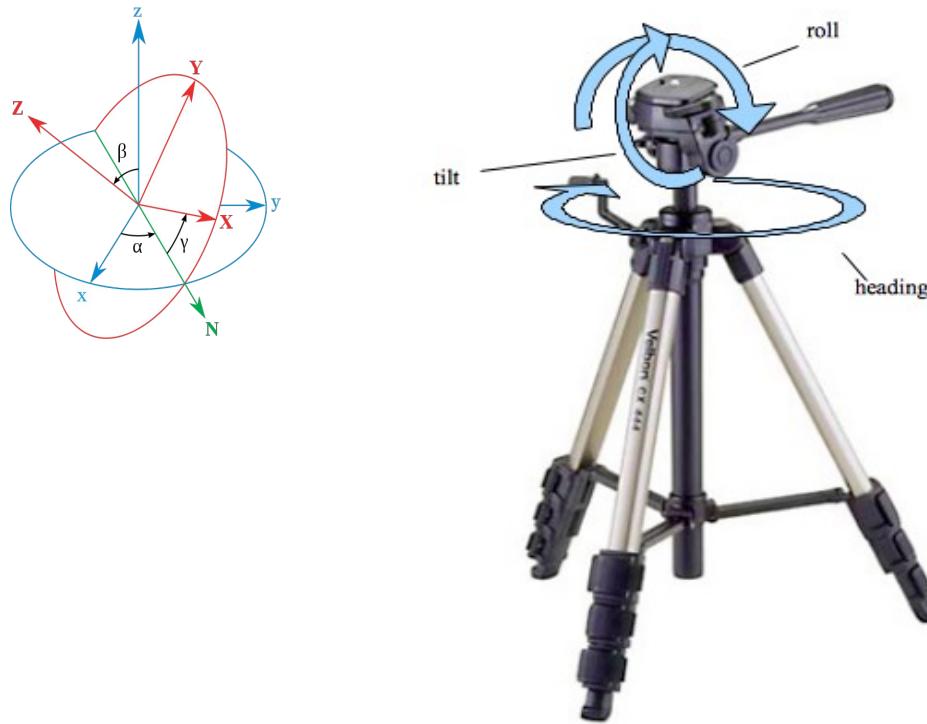


1.8.3 Sobre el movimiento en el trípode:

Conviene comentar que los movimientos que tratan de medirse son los que realiza el rig de cámaras sobre el trípode. Se ha elegido como punto de partida por su estabilidad y su empleo masivo. Además los movimientos son extrapolables a cualquier otra técnica de rodaje, simplemente añadiendo algunas piezas más al sistema.

Como se ve en la ilustración, todos los movimientos se realizan siguiendo los ejes polares. Por tanto el resultado final será la posición en grados respecto a los ejes X e Y (paralelos al suelo) y el cabeceo, medido en grados respecto del norte magnético.

El nombramiento de los ejes se hace siguiendo el criterio de Euler.



 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	---------------------------

Más allá del trípode, ya se está planeando la manera de medir en otras condiciones que puedan surgir en el rodaje. En el caso del desplazamiento sobre railes, se incluirá en el futuro un encoder en una de las ruedas para saber el desplazamiento y la velocidad.

1.8.4 Sobre el uso de Arduino [8]:

La plataforma Arduino ha sido fundamental en el desarrollo del conjunto de proyectos de la empresa. Por un lado, su sencillez y la ingente cantidad de ayuda que se puede encontrar en los foros hace que la curva de aprendizaje sea muy asequible. Por otro lado permite el uso de múltiples librerías compatibles, ya que es una plataforma muy utilizada en todo el mundo y la mayoría de los dispositivos electrónicos y software de programación ya facilitan el entrelazamiento con Arduino.

Téngase en cuenta también que una empresa en que entran y salen colaboradores de distintos campos exige el uso de una plataforma sencilla y genérica. Además se está investigando en campos tan dispares como los sensores y el control de motores. Un hardware único es el primer paso para un software común.

A pesar de lo dicho, no han podido localizarse librerías utilizables directamente en el sistema, con lo cual se ha tenido que implementar la mayor parte del software casi desde cero. La clase arduino.m para Matlab (de la que se hablará más adelante) si que partió de un aporte de la comunidad que permitía el manejo de Arduino desde el citado

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

programa. Sin embargo sólo se aprovecharon algunas de las funcionalidades incluidas, mientras que la mayor parte del código fue reprogramado.

1.9 Planificación.

1.9.1 Etapas.

- Seguimiento del proyecto.
- Búsqueda de información y toma de datos.
- Estudio de posibles soluciones.
- Realización de diseños preliminares.
- Preparación de la memoria.
- Selección y definición del diseño final.
- Búsqueda de componentes.
- Programación del software del sistema.
- Creación de un prototipo.
- Elaboración de los documentos básicos.
- Confección de planos.

1.9.2 Cronograma.

Tarea/Semana	1	2	3	4	5	6	7	8	9	10	11	12
Seguimiento del proyecto	x	x	x	x	x	x	x	x	x	x	x	x
Búsqueda de información y toma de datos		x	x	x	x	x	x	x				
Estudio de posibles soluciones	x	x										
Realización de diseños preliminares		x	x	x								
Selección y definición del diseño final				x	x	x	x	x				
Búsqueda de componentes			x	x								



Confección de planos								x	x	x	
Programación del software			x	x	x	x	x	x	x	x	x
Elaboración de la memoria				x	x	x	x	x	x	x	x
Creación de un prototipo		x	x						x	x	

Además de estos últimos tres meses de actividad, fueron necesarios otros tres para el aprendizaje de programación en C y los conceptos técnicos del cine 3D. Durante ese tiempo también se hizo parte del estudio de posibles soluciones.

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

2. ANALISIS DE SOLUCIONES

2.1 Decisiones de diseño

En este apartado de la memoria se indicarán las distintas alternativas estudiadas, los distintos caminos que se han seguido para llegar a ellas, las ventajas e inconvenientes de cada una de las mismas, y finalmente la solución adoptada y su justificación.

Entiéndase como un resumen de las decisiones más importantes tomadas, especialmente durante las primeras semanas del proyecto. Para ellas se necesitó el visto bueno de la empresa contratante. Es los apartados 2.2 y 2.3 donde se explica a fondo todo el desarrollo técnico del proyecto.

2.1.1 Tipos de movimiento y sensores

Lo primero que se hizo en el proyecto fue definir los movimientos que era necesario medir. Existen tres movimientos que puede realizar la cámara (en este caso el *rig²* de cámaras) sobre un trípode. Son *inclinación, giro y cabeceo*¹¹.

En primer lugar se analizó una solución de hardware abierto que se estaba utilizando para control de videojuegos. Se trataba de un mando inalámbrico que era capaz de detectar su posición absoluta en el espacio gracias a la un emisor de campo magnético. El mando interrumpía el campo, delatando su posición. Se tuvo que descartar

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

esta posibilidad ya que no se pudo contactar con el proveedor estadounidense del producto, a pesar de la insistencia de la empresa.

Después se barajaron las soluciones adoptadas por *Wiimote*¹² y *Kinect*¹³ pero resultaron inservibles por estar ideadas para el salón de casa. Perdían su funcionalidad en campo abierto debido a su dependencia de un sensor de infrarrojos. El alcance es sólo de unos 10 metros, y la luz del sol incidiendo directamente introduce un enorme ruido en toda la banda de frecuencias.

En última instancia se optó por diseñar un pequeño sistema con los sensores necesarios para la medición, así como un software que permitiera la monitorización en tiempo real. Se decidió hacer algo básico para empezar, que fuera funcional. Por otro lado mediante el uso de una placa de microcontrolador para la adquisición de datos se dispondría del número de pines necesario para posteriores modificaciones (como se especifica en el apartado 3.2).

Para realizar mediciones de la inclinación en el eje X y en el Y sólo es necesario un acelerómetro de tres ejes. Sin embargo el cabeceo comporta una mayor dificultad. En primera instancia se pensó en el uso de giroscopios, pero se vio que sólo median velocidad y no posición. La única manera de conocer el ángulo de cabeceo pasa por la implementación de una brújula electrónica.

El dispositivo se basa en un magnetómetro que mide el campo magnético del eje de la tierra y de este modo se indican los grados de desvío respecto al norte magnético.

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

Así pues de decidió que el sensor utilizaría acelerómetro y magnetómetro, ambos de tres ejes para cubrir todas las posibilidades.

2.1.2 Magnetómetro y acelerómetro

Al realizar la búsqueda de magnetómetros comerciales surgió una cuestión importante. El magnetómetro mide bien si se mantiene sin inclinación, pero en cualquier otro caso hay que medir dicha inclinación con un acelerómetro y aplicar una ecuación matricial para compensarla. Por este motivo nos decidimos por un modelo (LSM303DLH) que incluye magnetómetro y acelerómetro en un mismo integrado.

Más tarde se descubrió que tanto el magnetómetro como el acelerómetro exigían una compleja calibración, lo cual redundaba en la necesidad de hacer una buena interface de comunicación con Matlab.

2.1.3 Placa del sensor

Es decisión de la empresa 3+D entertainment el trabajar en la medida de lo posible con software y hardware abierto. En su opinión, este tipo de tecnología aporta fiabilidad por estar supervisada por la comunidad, así como de disponibilidad y alta tasa de actualización.

Otra de las razones es el ahorro de tiempo y costes, que puede resultar crítico en una empresa de nueva creación con base innovadora.

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria
		14/11/2011

Con la intención de entrar rápidamente en la programación del software, para lo que se necesitaba un prototipo, se decidió hacer uso de PCBs de código abierto que incluían lo necesario para el uso del integrado LSM303DLH.

Por un lado se puede adquirir la placa fabricada para el prototipo, por el otro se permite la descarga de los esquemáticos. De este modo se puede modificar según los intereses de cada usuario.

2.1.4 Placa de microcontrolador

Esta fue la decisión más sencilla, puesto que en buena medida venía exigida por la empresa por varias razones. En primer lugar, como se expresaba en el apartado anterior, se está tratando de apostar por plataformas abiertas de hardware y software. Por otro lado, los proyectos con motores ya estaban siendo realizados con Arduino, y además cumple con las especificaciones de presente proyecto.

Dentro de las opciones disponibles se ha escogido Arduino Duemilanove por ser la versión más genérica y se disponía de placas en desuso en la empresa. No obstante, ya que no se emplean demasiadas entradas/salidas no se descarta en el futuro el uso de Arduino Nano. Se trata de una versión más pequeña y económica, pero totalmente compatible.

2.1.5 Alimentación

En un principio se habló de las distintas posibilidades de

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	---------------------------

alimentación que había disponibles para la placa Arduino. Incluso se barajó la posibilidad de hacer un sistema inalámbrico por bluetooth, pero se vio que carecía de sentido práctico.

La razón es que el proyecto forma parte un sistema mayor que incluye visión por computador y control de motores, por lo que tiene que haber un ordenador cercano. Esto significa que se puede mantener la alimentación por USB. Además siempre debería estar conectado a un ordenador portátil para la transferencia de datos, permaneciendo alimentado todo el tiempo.

2.1.6 Arduino o Matlab

La decisión más costosa y que más que más variaciones a sufrido a lo largo del proyecto es que parte del software estaría en Matlab y cual en Arduino. Habiendo hecho una profunda interface que permitía un gran control desde Matlab, aparecen varias opciones opciones.

Los dos casos extremos serían:

1- Tratar de ejecutar *medidas I2C, calibración de sensores, compensación de inclinación y filtros de señal por software* en el Arduino. Usar Matlab únicamente para representación gráfica. Esta es la opción ideal, pero poco realista.

2- Usar Arduino como tarjeta de adquisición de datos y hacer todos los cálculos en Matlab.

Se ha elegido una opción intermedia, en la que Arduino realiza las

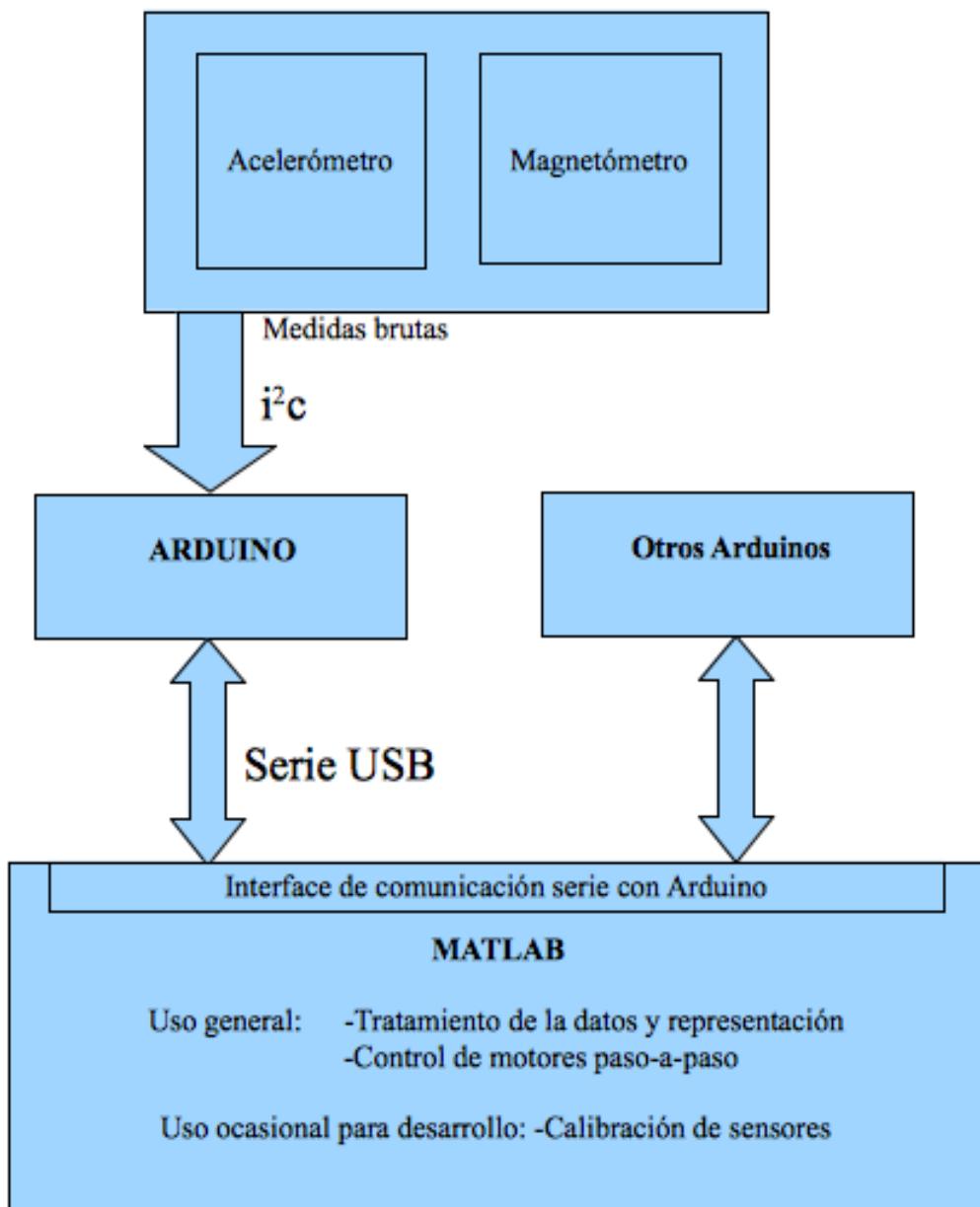
 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

medidas I2C, y un primer filtro “paso bajo” por software. La calibración y la compensación son mucho más eficientes en Matlab porque se hacen con ecuaciones matriciales. Esto libera a Arduino y le permite ejecutar más medidas y transferencias serie por segundo. De este modo obtenemos una cantidad suficiente de datos para trazar trayectorias, hacer representaciones gráficas, etc. Además se incluyó en última instancia un filtro en Matlab previo a la representación para suavizar las medidas.

Se puede encontrar una extensa explicación del tema en la sección 2.3 referente a la descripción del software.



2.1.6 Representación esquemática de la solución



 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

2.2 Descripción y documentación del hardware.

En este apartado se va a entrar en detalle con los componentes del sistema y los tipos de conexión que existen entre los tres bloques principales: sensor, placa de microcontrolador y ordenador.

2.2.1 Cómo funciona un acelerómetro.

Como su propio nombre indica, el acelerómetro es un instrumento destinado a medir aceleraciones. No obstante, no mide lo que intuitivamente se entiende por aceleración (variación de la velocidad en el espacio) sino la asociada al peso de una masa de referencia. Esto quiere decir que si se mide en el eje de un acelerómetro orientado hacia el suelo, la lectura debería ser equivalente a 1g, la aceleración de la gravedad.

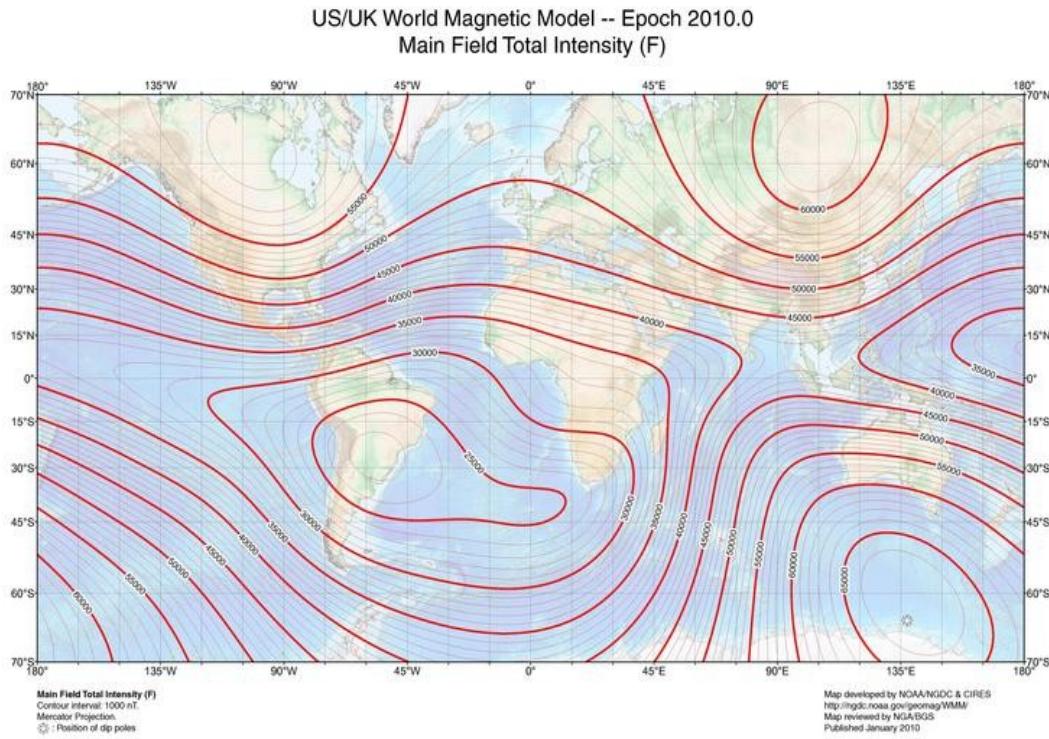
Los acelerómetros más modernos ya no funcionan con una masa, puesto que eso impediría alcanzar tamaños propios del montaje superficial (SMD). La tecnología que utilizan se basa en el traspaso térmico por convección natural, pudiendo medir la transferencia de calor causada por la aceleración.

2.2.2 Cómo funciona un magnetómetro.



Los magnetómetros son aparatos capaces de medir la fuerza o dirección de un campo magnético. El utilizado en este proyecto sólo es capaz de medir la fuerza.

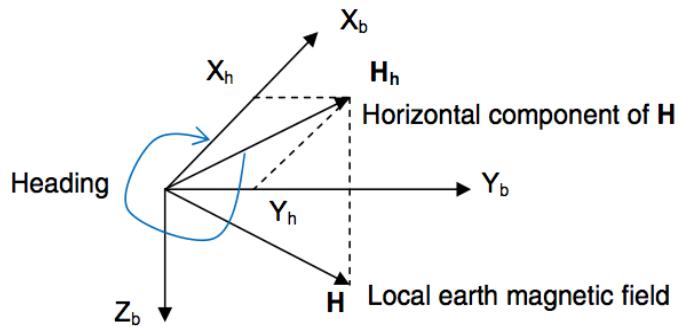
El principio de funcionamiento del sensor es captar el campo magnético de la tierra [4], que es constante en el tiempo y el espacio mientras no haya un gran cambio de latitud. Para ilustrar la situación se adjunta una imagen con la distribución de la intensidad de campo magnético terrestre. (las medidas medidas están en Teslas, 1gauss=10.000T)



En base a esto se puede saber que posición ocupa el sensor respecto de dicho campo magnético, creando así una aplicación de



brújula electrónica.



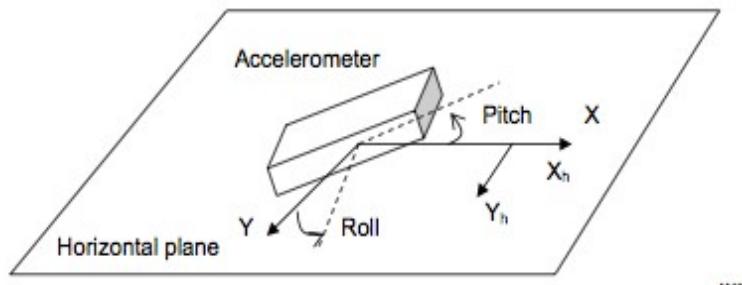
El campo magnético local de la Tierra (H) tiene una componente H_h fija en el plano horizontal (x, y) que apunta al norte magnético. Esta componente puede ser medida por el sensor a través de las componentes X_h e Y_h , que se encuentran en los ejes de medida del magnetómetro.

La ecuación a utilizar para calcular el cabeceo es la siguiente:

$$\text{Heading} = \arctan(Y_h / X_h)$$

Hay que tener en cuenta que para la correcta medida del campo magnético, los ejes del magnetómetro deben coincidir con los del susodicho campo. Estos ejes son el perpendicular al eje de la tierra (z) y los paralelos al suelo (x e y).

Si el sensor se inclinara, habría que tener en cuenta el dato de la inclinación para compensar la medida. Esto se debe a que el módulo del vector resultante (fuerza magnética medida) sería menor.



Para ello sería necesario aplicar la siguiente ecuación:

$$X_h = X_M \cos \text{Pitch} + Z_M \sin \text{Pitch}$$

$$Y_h = X_M \sin \text{Roll} \sin \text{Pitch} + Y_M \cos \text{Roll} - Z_M \sin \text{Roll} \cos \text{Pitch}$$

Donde:

- X_h e Y_h son los valores corregidos,
- pitch* y *roll* son inclinación y giro (en castellano son nombres algo confusos por no estar normalizados).
- X_M , Y_M y Z_M son los valores medidos por el sensor.

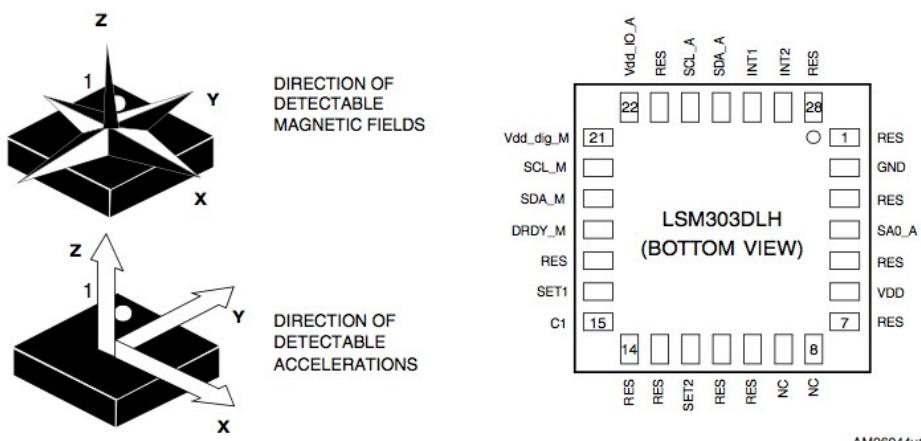
2.2.3 La placa del sensor

El corazón de la placa es un integrado de la casa STMicroelectronics. Se trata de un acelerómetro y un magnetómetro, ambos de tres ejes. El segundo depende del primero para su correcto funcionamiento, pero además del conjunto se han utilizado las medidas del acelerómetro como sensor independiente.

El integrado es el LSM303DLH [11] y cuenta con una serie

características que se detallan a continuación:

- Tensión de alimentación analógica de entre 2.5 y 3.3v
 - Tensión de alimentación digital de 1.8v
 - Modo de bajo consumo
 - 3 canales de campo magnético y 3 de aceleración
 - Escala de campo magnético entre 1,3 y 8,1 gauss
 - Escala de aceleración de ± 2 g/ ± 4 g/ ± 8 g dinámicamente seleccionable.
 - Salida digital de 16 bit.
 - 2 generadores programables de interrupción para detección de caída libre.
 - Interface serie I²C
 - Función de hibernación para el acelerómetro
 - Detección de orientación en 6 direcciones
 - ECOPACK ajustado a la normativa RoHS



Ejes de los sensores y esquema de pines del integrado.

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

A continuación se detallan las cuestiones y decisiones relativas a las diferentes características del integrado LSM303DLH:

Tensiones de alimentación:

Para lograr estos requisitos se ha optado por alimentar la placa desde Arduino que proporciona 3.3 o 5v. De las dos, se ha elegido la de 3.3 por se la más próxima a las necesarias para el integrado. No obstante ha sido necesario añadir sendos reguladores de 1.8 y 3.3v para alcanzar las tensiones de alimentación digital y analógica respectivamente.

Técnicamente hubiera sido suficiente con el regulador de 1.8, pero de este modo es posible la conexión con tensiones de alimentación mayores y se logra un grado más de protección frente a sobretensiones y sobrecorrientes.

Cambios de escala:

La escala en que se toman las medidas puede ser modificada en ambos sensores. Concretamente en el acelerómetro se permite realizar esa modificación dinámicamente, durante la ejecución de la lectura de medidas.

Para el propósito de este proyecto interesa medir a una escala que permita la captura de movimientos suaves. Para el acelerómetro, en ningún caso deberíamos registrar aceleraciones superiores a 2g o

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	---------------------------

incluso menos. Por su parte, el magnetómetro mide la fuerza del campo magnético de la Tierra, que es constante y del orden de 0.4 gauss en España. La escala de 1.3 gauss es la más adecuada.

Por defecto, el integrado viene con esta configuración, pensada para la mayor parte de las aplicaciones típicas de este sensor. No obstante dispone de unos registros para que pueda modificarse a gusto del usuario.

Interface serie I²C:

Las lecturas son transmitidas al microcontrolador mediante el protocolo I²C que permite conectar múltiples sensores en paralelo y llamar al que más interese. En este caso sólo hay uno, aunque no se descarta en el futuro añadir un segundo sensor idéntico, pero en posición diametralmente opuesta, para obtener redundancia.

Las medidas son recogidas por el integrado en 12 registros independientes (2 por cada medida, para parte alta y baja del dato). Hacen un total 6 medidas pertenecientes a los ejes x , y , z del acelerómetro y lo mismo para el magnetómetro. Así se consigue el propósito de medir cabeceo, compensando el magnetómetro con la inclinación, y también aprovechar el acelerómetro como medida independiente de inclinación en los tres ejes.

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

Funciones extra:

Las funciones de reducción de consumo como la de hibernación del acelerómetro han sido obviadas puesto que son irrelevantes para esta aplicación. El dispositivo está siempre alimentado y el consumo es minúsculo (0.83mA).

Las interrupciones para detección de caída libre no han sido implementadas porque carecían de sentido para el propósito del prototipo. No obstante en el futuro podrían tener cierta utilidad a la hora de activar rutinas de interrupción en caso de cambios bruscos.

Otras características relevantes:

- Rango de temperaturas entre -30 y 85°C.
- Sensibilidad de aceleración lineal 1 mg/dígito. Para 12 bit, 4096 dígitos. (1 LSB= $4\text{g}/4096 = 0.001\text{ g}$).
- Resolución magnética típica de 8 m gauss. (1 LSB= $1,3\text{ gauss}/4096 = 0.0003\text{ gauss}$)

El rango de temperaturas permite su empleo en situaciones climáticas extremas. La resolución de sus medidas también es en principio suficiente para los propósitos de este proyecto.

En cualquier caso hay que apuntar que ha aparecido ruido en las medidas cuyo origen no viene especificado en la hoja de datos del fabricante. Este problema se ha paliado en buena medida mediante la implementación de filtros paso bajo por software.

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

Cálculo del ruido en las medidas:

Es necesario hacer un cálculo para saber la precisión real de los sensores en cuanto a magnitudes físicas, después se calculará la influencia real del ruido.

El nivel de ruido ha sido recogido de forma empírica mediante Matlab.

Acelerómetro: ruido de fondo ± 6 LSB aprox.

$1 \text{ LSB} = 0.001\text{g}$ Inclinación máxima $\pm 90^\circ$

En el peor de los casos tendríamos una variación de 2g (de -1g a 1g) para una variación de 180° volcando por completo el eje en que se está midiendo. $2\text{g}/180^\circ = 0,011\text{g/grado} \Rightarrow$ unos 10 LSB por grado.

El ruido es del orden de 1 grado, en algunos casos incluso puede haber variaciones de $\pm 1^\circ$. Obviamente hay que recurrir a filtrado.

Magnetómetro: ruido de fondo ± 10 LSB aprox.

$1 \text{ LSB} = 0,0003\text{gauss}$ Cabeceo máximo $\pm 180^\circ$

El cálculo es similar al caso anterior. El campo magnético terrestre es del orden de 0.45gauss en España. La variación en un giro de 180° sería de $0,45$ a $-0,45$.

$0.9\text{gauss}/180^\circ = 0,005\text{gauss/grado} \Rightarrow$ unos 16 LSB por grado.

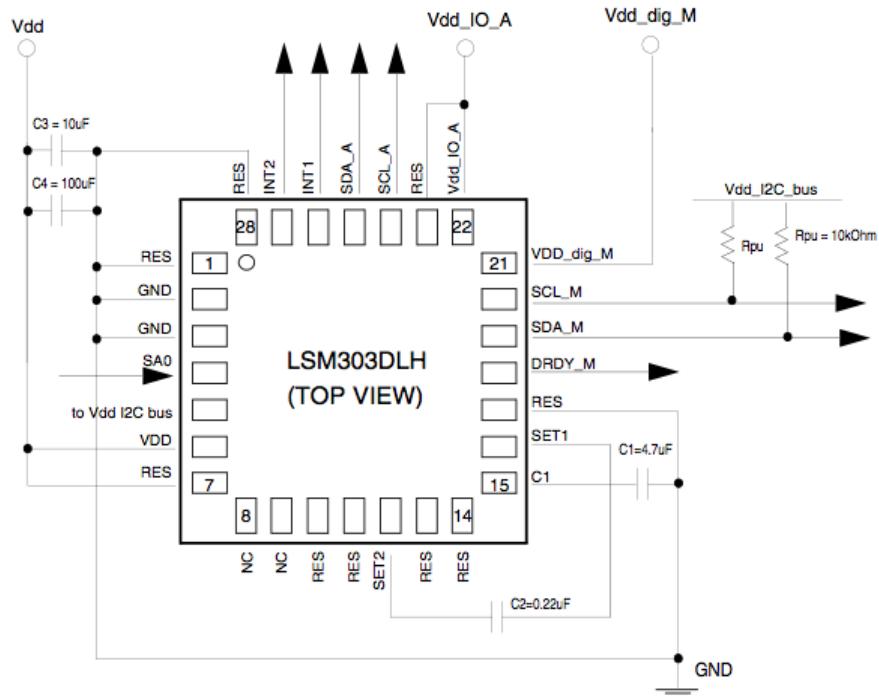
El error introducido por el ruido es mayor de 1 grado, por lo que se hace imprescindible filtrar para obtener medidas fidedignas.

Otros componentes de la placa:

El fabricante del integrado hace una serie de recomendaciones relativas al hardware adicional que necesita para funcionar. En definitiva deja el asunto prácticamente solventado, de tal manera que sólo resta situar los componentes en una PCB para disponer del sistema funcionando.

Se ha observado que los diferentes fabricantes que venden la placa completa como hardware abierto han seguido al pie de la letra las indicaciones dadas en el datasheet.

A continuación el esquema que se muestra en la hoja de datos y aclaraciones al respecto:



Conexiones recomendadas por el fabricante

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

Capacidades externas

Los condensadores C1 y C2 deben ser de tipo cerámico. El primero tiene su propio pin reservado y debe ser estrictamente de 4,7uF. C2 es el filtro para las señales internas de Set/Reset. Su valor debe ser de 0.22uF.

Los condensadores C3 y C4 son los filtros para alimentación. Como tales deben estar lo más cerca posible del integrado para eliminar los efectos de inductancias parásitas. Se recomienda que el primero sea electrolítico de aluminio, de valor 10uF. C4 debe ser cerámico de 100nF.

Resistencias Pull-up:

Con la intención de asegurar el correcto funcionamiento de la transmisión de datos I2C, el fabricante recomienda el uso de resistencias de pull-up en las líneas SDA y SCL. Se encargan respectivamente de la transmisión de datos y de señal de reloj.

Las resistencias deben tener valores medianamente elevados. El fabricante recomienda 10k aunque podrían ser mayores.

Alimentaciones:

Como se ha dicho antes, la alimentación analógica y la digital son distintas (3.3 y 1.8v respectivamente). Los pines a conectar a la tensión analógica son el 6 y 7. Para los 1.8v digitales irán el 5, 21, 22 y 23. A masa se conectan 1, 2, 3, 17 y 28. La masa es compartida con la

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

placa Arduino.

Otros:

El integrado incluye una opción llamada *configuración SA0* que consiste en la inclusión de un pin que se corresponde con el bit menos significativo de la dirección I2C del sensor. Si ese pin se deja a 0, se podría incluir otro integrado igual en paralelo. Dicho integrado extra debería tener el SA0 a 1 y de este modo tener una dirección I2C distinta.

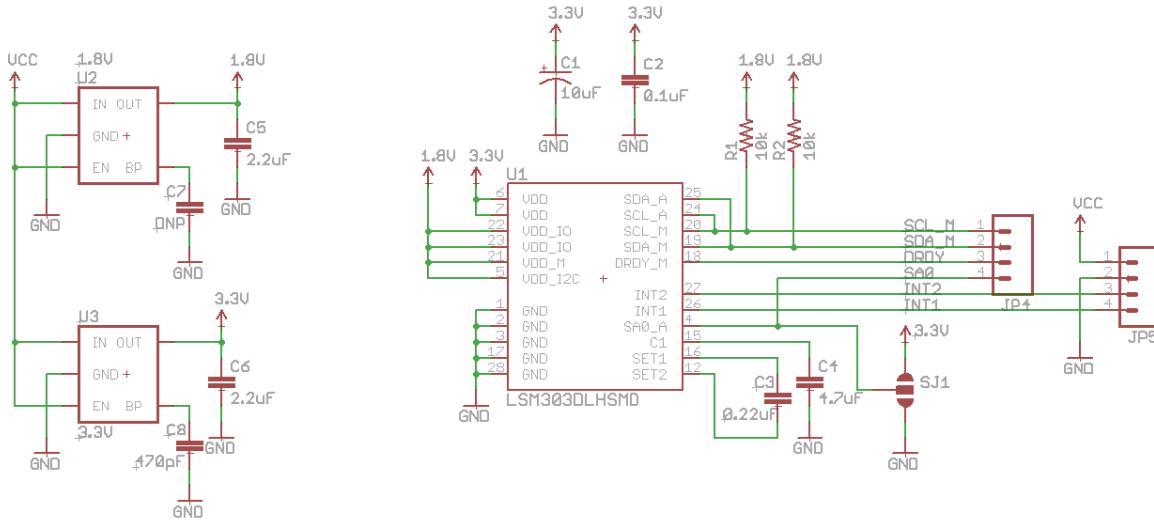
Para ello se incluye en la PCB un selector entre 3.3v y masa que se debe seleccionar mediante soldadura con una u otra opción.

Reguladores de tensión: [14]

Para los reguladores de tensión es recomendable algún LDO, reguladores con pequeña tensión de drop-out. Se puede elegir entre muchos fabricantes, Sparkfun recomienda para esta aplicación el empleo de los MIC5205 de Micrel.

Para el consumo de corriente realizado (0,83mA), la Vdrop del regulador es de unos 20mV. Si alimentamos a 3.3v desde Arduino, llegarán en el peor de los casos 3.28v al integrado.

Para el modo “bajo ruido” de los reguladores, el fabricante recomienda un condensador en el pin 4 de 470pF. En el pin 5, la salida, recomienda uno mayor de 2.2uF.



Esquemático de las conexiones del LSM303DLH y reguladores de tensión.

Nota: Véase que la numeración de condensadores no coincide con la del datasheet.

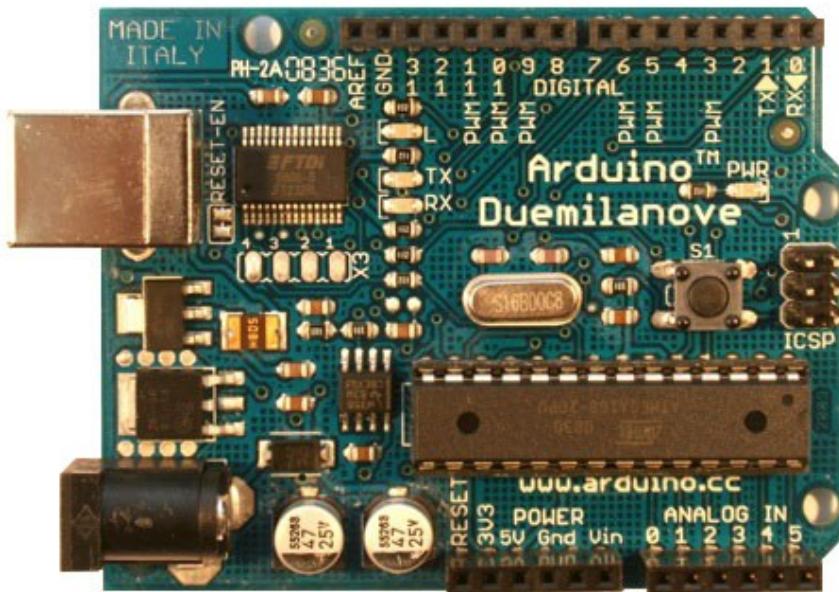
2.2.4 La placa Arduino.

Como se ha explicado anteriormente, se ha empleado Arduino Duemilanove en buena medida porque es la placa que ya estaba siendo empleada en otros proyectos de la empresa contratante, que a su vez están relacionados con este. Esto a largo plazo permitirá la unificación de todos ellos con un software común y protocolos de transmisión de datos similares, en la medida de lo posible.

Aunque dichos protocolos no son estrictamente iguales, puesto que unos son para motores y otros para sensores, bien es cierto que comparten características comunes. Especialmente la estructura y el código necesarios son muy similares. Se profundizará en ello más



adelante.



Fotografía de la placa Arduino Duemilanove

Las características básicas de Arduino Duemilanove son las siguientes:

Microcontrolador:	Atmega368 [13]
Voltaje de funcionamiento:	5V
Voltaje de entrada (recomendado):	7-12V
Voltaje de entrada (límite):	6-20V
Pines E/S digitales:	14 (6 proporcionan salida PWM)
Pines de entrada analógica:	6
Intensidad por pin:	40 mA
Intensidad en pin 3.3V:	50 mA
Memoria Flash:	32 KB (ATmega328) de las cuales 2 KB las usa el gestor de arranque (bootloader)

 1542	Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria
			14/11/2011

SRAM:	2 KB (ATmega328)
EEPROM:	1 KB (ATmega328)
Velocidad de reloj:	16 MHz

En general cumplen ampliamente con las especificaciones del proyecto. Sólo se están usando los pines analógicos 4 y 5. Son los que se ocupan del I2C, se corresponden con la transmisión de datos (SDA) y el reloj (SCL) respectivamente. Aparte de esto, se alimenta el sensor con la salida 3V3 que proporciona 3.3v, y unen la masas de ambas placas a través del pin GND.

La placa de Arduino es alimentada vía USB al mismo tiempo que se comunica con el ordenador. También se podría alimentar con un alimentador externo de hasta 12v CC. Tiene reguladores internos de 5 y 3.3v para adaptar la tensión de alimentación a las necesarias por los componentes de la placa.

Para más detalle, la información completa del dispositivo se encuentra en la siguiente dirección web:
<http://arduino.cc/es/Main/ArduinoBoardDuemilanove>

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

2.3 Descripción y documentación del software.

2.3.1 Introducción y esquema.

El grueso del trabajo y el tiempo dedicado a la puesta en marcha de este sistema reside en el software. En los próximos apartados se detallarán los pasos seguidos en su desarrollo, los problemas encontrados y como se han solucionado.

Esta es la lista de programas y funciones implementadas, en orden de aparición:

- Librería de lectura de datos mediante protocolo I2C entre Arduino y sensores.

- Sketch¹⁸ de la placa Arduino para capturar los datos del sensor, tratarlos y enviarlos por puerto serie a Matlab.

- Librería para control de placas Arduino mediante puerto serie desde Matlab (en adelante clase arduino.m). Sirve tanto para este proyecto como para control de motores.

- Diversas funciones de filtrado de datos en tiempo real, para incluir en el sketch de Arduino.

- Funciones de calibración del sensor para Matlab. Incluye filtrado digital, calibrado del acelerómetro, eliminación de efectos "hard-iron" del magnetómetro y corrección de alineación.

- Función de cálculo de inclinación.

- Función de cálculo de cabeceo ("heading") que además corrige el

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

error en la medida provocado por la inclinación.

-Representación gráfica en tiempo real.

2.3.2 Librería de lectura de datos mediante protocolo I2C entre Arduino y sensores. [9]

El integrado LSM303DLH guarda las medidas realizadas en unos registros determinados, que son los que hay que leer desde Arduino. Esta mini-librería abstrae al programador del *sketch* de este proceso haciendo la lectura de los 12 registros. En ellos esta guardada la información en forma “little indians”, con parte baja y alta de palabra. Al guardar los datos en Arduino se reconstruyen los datos dejándolos en forma de 6 valores decimales con signo.

La librería está escrita en lenguaje C++ y se integra como “include” en el sketch de Arduino. Además se ha creado también una clase llamada *compass* (brújula en inglés) donde se guardan siempre los 6 valores para su envío a Matlab.

Hay que añadir que esta librería se combina con la propia interna de Arduino para conexiones I2C (llamada *Wire*), la cual tiene embebido el código para conectar y desconectar con el sensor.

2.3.3 Sketch de la placa Arduino

El programa que ejecuta Arduino está organizado de la siguiente manera. Por un lado el bucle principal se dedica a leer y almacenar en un array los datos del sensor. Cada vez que se comienza este proceso

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

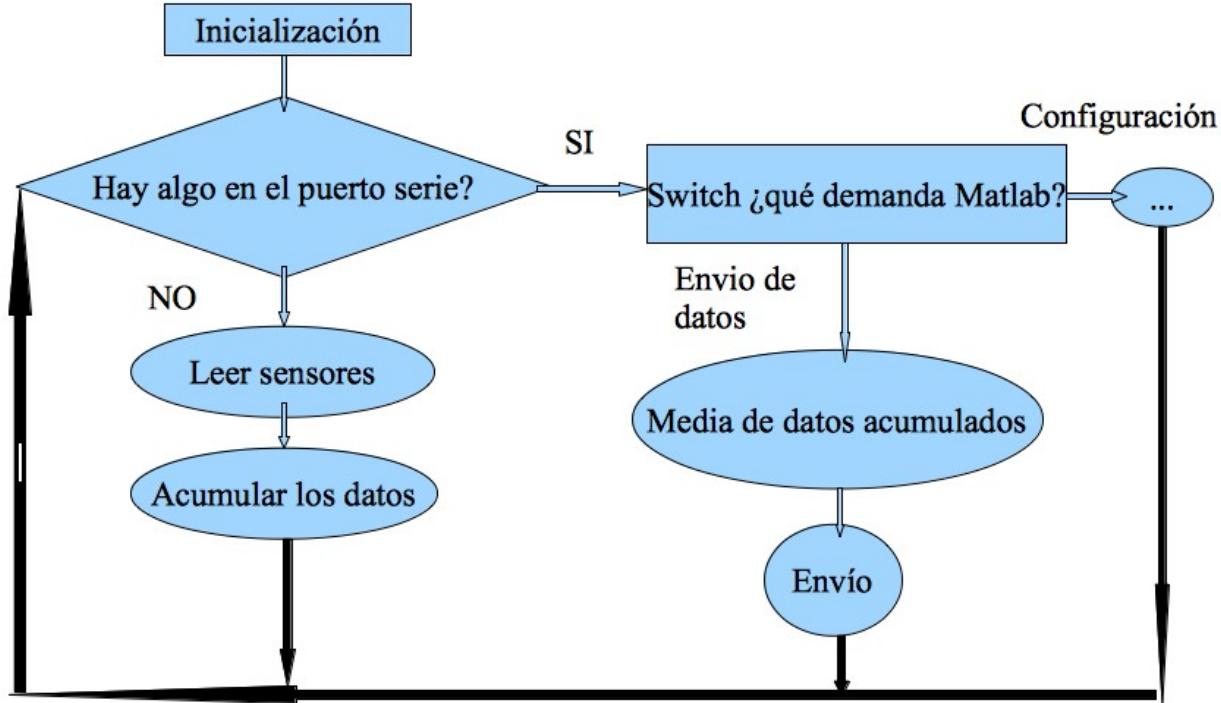
se comprueba el puerto serie para ver si hay una petición por parte de Matlab, en cuyo caso se atiende.

En la versión final lo único que se recibe por puerto serie es la petición de envío de datos a Matlab. Sin embargo hay inhabilitadas otras opciones que permiten configuraciones remotas, como cambiar el tipo de filtrado que Arduino hace a los datos antes de enviar (se implementó para poder probar los diversos filtros y medir su efectividad desde Matlab, comparando los resultados).

Una vez que se confirma que hay una petición en el puerto serie se pasa a ejecutar el filtro pertinente. En la última versión es una media con diezmado de datos. Después de filtrar se envían los datos según el protocolo que implementamos en Matlab para que éste los pueda entender.

El protocolo de envío de datos se explica detalladamente en la sección 2.3.4.

Conviene resaltar que los motores son controlados por otra placa Arduino con su propio sketch implementado paralelamente por otros compañeros de la empresa. El modo de funcionamiento es muy similar ya que el esquema a seguir se decidió de mutuo acuerdo.



Filtros con media [5]

A continuación se trata el tema de los filtros de media para los datos. Se trató de buscar un buen filtro que actuara a modo de paso-bajo con los datos, ya que había un ruido importante en las medidas. La idea era neutralizar este ruido dentro del sketch de Arduino, ya que en teoría se pueden hacer muchas medidas por I2C entre petición y petición serie de Matlab. De este modo se trabaja de manera muy similar a como lo haría una interrupción.

Empíricamente los resultados fueron bastante peor de lo previsto. Al tener que usar matrices de 6 datos para todo el tratamiento, los ciclos de reloj y la memoria ocupada se disparaban. Explicado punto

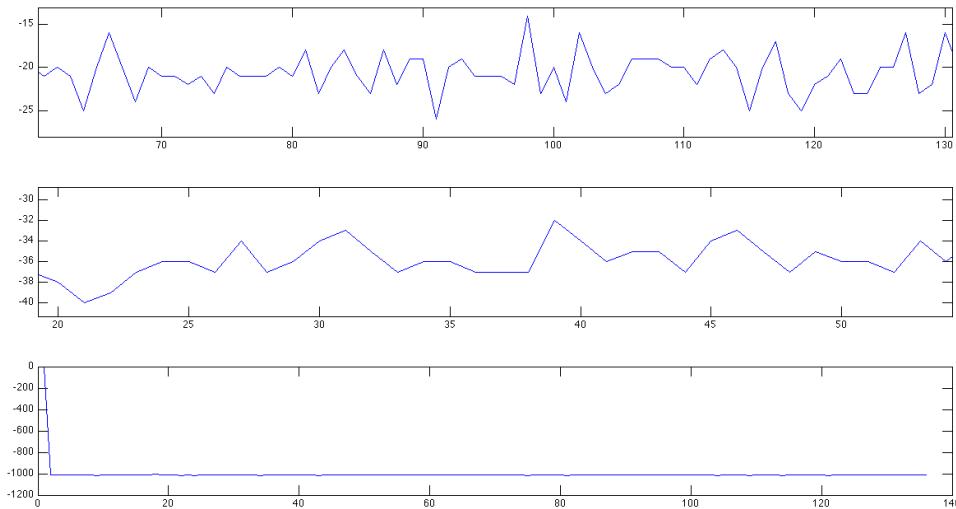


por punto:

-Medidas brutas: Cogiendo una medida cada vez que hay una petición serie. Es muy rápida pero con demasiado ruido. Concretamente el tiempo consumido en Arduino es despreciable frente a la gestión en Matlab, mientras que el error llega hasta 12 LSB (+-6) en situación de estabilidad total.

A continuación una muestra del perfil de ruido del acelerómetro. Se tomó como referencia por defecto en lugar del magnetómetro, aunque el ruido y el efecto de los filtros es equivalente en ambos.

El eje vertical son LSB y el horizontal segundos. El sensor estaba en completo reposo durante la medida.



-Media simple: realizaba la media de las últimas n medidas antes de la petición serie. Funcionaba bien a partir de 100, logrando errores

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

de 2 LSB (+-1) prácticamente insuperables. Sin embargo se creaba un cuello de botella en Arduino. El coste en tiempo era de 200ms, permitiendo en total recoger 4 datos por segundo desde Matlab. Obviamente fue descartada.

-Media acumulativa: En vista de que no se estaba trabajando con todos los datos disponibles, se hizo un algoritmo que lo permitiera. Se trata de una media acumulativa, escrita de la siguiente forma:

$$\text{media_actual} = \text{media_anterior} + ((\text{nuevoValor} - \text{media_anterior}) / (n+1))$$

Desafortunadamente seguían persistiendo el problema de tiempo de ejecución en Arduino. Si se reducía era a costa de que la conexión serie (durante la que no se recogen medidas) fuera casi tan larga como el tiempo en que Arduino trabajaba en hacer lecturas del sensor.

Empíricamente se vio que no aportaba gran precisión, unos 8 LSB a costa de recoger menos de 10 medidas por segundo.

-Media diezmada: Por los resultados se dedujo que merecía más la pena ser menos ambiciosos, cogiendo unas pocas medidas separadas en el tiempo. Luego, al recibir la petición serie se realizaría la media muy rápidamente.

Se midió cual era la velocidad real de la petición serie desde Matlab usando el “Profiler”, una utilidad que mide el gasto de tiempo de las funciones durante la ejecución. El resultado es de unos 14ms,

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	---------------------------

aunque puede variar un poco dependiendo del ordenador.

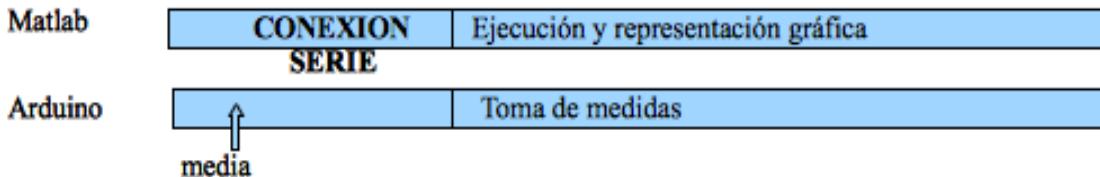


Gráfico de tiempos de trabajo de Arduino y Matlab

A continuación Matlab procede a realizar el “plot” para la representación gráfica. Durante ese tiempo (alrededor de 24 ms) Arduino va tomando medidas cada 1ms y almacenándolas. Luego Matlab llevará a cabo otra petición serie y recibirá la media de las medidas tomadas por Arduino.

El error frente a las medidas brutas se rebajaba desde 12 hasta 5 LSB. La velocidad de lectura asciende a unas muy aceptables 25 medidas por segundo, que permite la representación en tiempo real que pretendíamos.

Con este sistema hay cierta pérdida de datos, pero no es muy relevante para ésta aplicación. En definitiva se están tomando alrededor de 25 medidas por segundo de un sensor. Sin embargo en vez de ser datos instantáneos, se trata de medias de unos cuantos datos consecutivos, reduciendo el ruido.

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

2.3.4 Librería para control de placas Arduino (*arduino.m*)

[10] Se podría decir que *arduino.m* es el cimiento de la mayor parte del proyecto puesto que todo el trabajo de Matlab se basa en esta clase. Originalmente era un código abierto encontrado en internet que básicamente pretendía utilizar Arduino de forma absolutamente independiente desde Matlab.

Para lograr este propósito había por un lado un sketch de Arduino muy extenso para realizar todas las funciones disponibles (I/O analógicas y digitales, PWM, control de servos, etc). Por el otro, *arduino.m* aglutinaba en casi 1000 líneas de código toda una interface de funciones de comunicación con la placa. Esto quiere decir que se podía ejecutar cualquier código típico de un sistema embebido desde Matlab en tiempo real.

A pesar de la gran funcionalidad que tenía, la complejidad del código era tremenda. Además no era útil para ejecutar código en Arduino, ya que lo ponía realmente difícil. Resultaba inviable modificar los detalles necesarios para cada pequeña aplicación.

Se decidió conservar la idea y la parte del código más básica. Luego se crearon dos protocolos mucho más sencillos de los existentes. Uno de ellos sirve para comunicarse con las placas que alberguen sensores, el otro con las que controlan los motores paso a paso. De este modo se dispone de un código mucho más limpio e intuitivo, que permite intervenciones y ampliaciones posteriores, a

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

diferencia del *arduino.m* original.

El código está estructurado de la siguiente manera:

Función constructor del objeto arduino:

```
function a=arduino(comPort)
```

Principalmente nos abstrae de la conexión entre Matlab y el puerto serie. Su función interna más importante es “aser” (abreviación de Arduino serial connection). “Aser” Permite la configuración de la velocidad de conexión, 115200 baudios por defecto, así como elegir el puerto al que se quiere conectar, que se le pasa a través de la función principal. La nomenclatura del puerto cambia entre arquitectura PC y Unix pero funcionan ambas.

Esta parte del código incluye también varias funciones de seguridad y avisos de errores, así como el destructor del objeto.

Función para mover motores:

```
function moverMotor(a,mot_ID, dir, ussteps, delay)
```

Permite una abstracción de modo que en una sola linea de código de pueda mandar toda la información que necesita Arduino para mover los motores paso a paso de las cámaras.

Envía los siguientes datos:

-Nombre del Arduino, ya que se pueden manejar varios objetos de este tipo desde el mismo programa. Típicamente “a” o “b”.

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

-Número del motor (del 1 al 4). Cada placa controla 4 motores. Dos de ellos producen en las cámaras desplazamientos longitudinales y los otros circulares.

-Dirección. Puede ser a derechas o izquierdas (1 o 2).
 -Número de pasos a mover. Cada motor paso-a-paso dispone de de 10000.

-Delay entre pasos. Esto sirve para que no se muevan de uno en uno sino en grupos más grandes. Por ejemplo un grado de giro serían 80 pasos. Con Delay=80 cada paso en la variable anterior sería un grado.

Funciones para leer información del Arduino:

```

function
[mot_1,mot_2,mot_3,mot_4,tiempo,ArduinoFree]=SerialRead(a)
function
[cabecera,ax,ay,az,mx,my,mz,t_readsensor]=SerialReadSensor(a)
  
```

Ambas son muy parecidas en su estructura. La primera sirve para leer la placa controladora de motores mientras que la otra es para la placa lectora de sensores. Voy a centrarme en explicar la segunda por ser ligeramente más completa.

En primer lugar envía un código (40) por puerto serie para indicar a Arduino que le tiene que mandar la información de los sensores. Otros códigos consecutivos como 41 o 42 se reservan para otras peticiones, como se verá más tarde. A su vez 10, 20 o 30 se han dejado libres pensando en la placa de motores y otras posibles

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	---------------------------

ampliaciones del sistema.

Inmediatamente después se chequea el puerto serie en busca de respuesta. Los datos llegan en forma de string de caracteres, separados por espacios. Las medidas pueden ir desde -1024 hasta 1024, pero hay que transformarlos de string a numérico.

Se inicializa un contador que sirve para recorrer todo el string. Cada vez que se llega a un espacio es que ha terminado un dato. Se convierte a decimal, se almacena y se continúa leyendo.

En este caso devuelve un 40, que es la misma cabecera, los valores del sensor y el tiempo de lectura. Este último está desactivado por defecto salvo que se quiera utilizar por algo en concreto.

Nota sobre las peticiones serie más complejas:

Como se explica en el apartado 2.3.3, durante el periodo de pruebas se desarrolló un sketch que permitía seleccionar entre diversos filtros y configurar algunos parámetros. Asimismo en el sketch para motores también se realizan peticiones serie con varios parámetros distintos.

La lógica que se sigue en el protocolo de comunicación es estructurar la trama con una cabecera de dos dígitos (4n en el caso del Arduino de sensores) seguido de los otros datos en un orden determinado. Si algunos de esos datos tienen un número de dígitos variable hay que emplear es siguiente procedimiento:

-El primer dígito indica el número de dígitos que tiene el dato. Se guarda ese número en una variable numDigitos.

-Realizar un bucle en el que se lea el siguiente dígito y se multiplique por el múltiplo de 10 que corresponda para que el número final sea el dato deseado. El bucle se repite tantas veces como indique numDigitos.

De este modo si queremos enviar 143 tendríamos lo siguiente:

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

41(cabecera)3(numero de digitos)143

El 1 se multiplica x 100, el 4 x10 y el 3x1 y con un bucle "for" se van guardando en la misma variable.

Este proceso se hace por razones de escalabilidad, siendo exportable a casos con una variedad cualquiera de dígitos.

Funciones de setup para la placa de sensores:

```
function setupNumMedidas(a, numMedidas)
```

```
function setupTipoPromediado(a, tipoPromedio)
```

Estas funciones se implementaron para poder configurar las medias que se hacían en Arduino. Una vez elegida una, han quedado inservibles. No obstante se van a conservar ya que se necesitan funciones similares siempre que se quieran hacer configuraciones remotamente.

Función para marcar posiciones iniciales en motor

```
function[mot_1,mot_2,mot_3,mot_4,tiempo,ArduinoFree]=ResetePosicion(a)
```

Se trata de una función muy sencilla para inicializar los valores de las variables que indican la posición de los motores. Se implementó en su momento cuando se probaba el uso simultáneo del sensor y los motores.

Primero se pone el prototipo de sensor sobre el motor y se ponen los valores de posición del motor a cero. Se le dice que se mueva un determinado número de pasos, se miran los datos del sensor, se pide

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

que vuelve al origen y así sucesivamente. De este modo podemos comprobar de manera precisa si el sistema de sensores tiene buena repetibilidad.

A pesar de que fue una prueba muy prematura, previa al filtrado y calibración, los resultados fueron satisfactorios.

2.3.5 Funciones de calibración de sensores en Matlab. [12]

Aunque en teoría los componentes vienen calibrados de fábrica, es bastante probable encontrar ciertos errores de medida. Se debe distinguir en primer lugar entre la calibración del acelerómetro y la del magnetómetro. La primera de ellas es conceptualmente más sencilla que la segunda.

A- Calibración del acelerómetro:

Explicación teórica.

Los errores que pueden aparecer en el acelerómetro y es necesario corregir son los siguientes:

-Mala alineación de los ejes del acelerómetro. *Matriz A_m*.

-Offset producido por errores constantes del sensor. *Matriz A_OSi*.

-Corrección del factor de escala de los ejes. Basicamente consiste en conocer los valores máximos de cada uno de ellos, para poder normalizarlos a una medida común (+-1). *Matriz A_SCi*.

Expresión matricial.



Aquí se ve cual es la expresión matricial que se está aplicando a la hora de ordenar las funciones. Como vemos, a las medidas crudas se les resta el offset, luego se multiplica por la matriz de alineación y a continuación se corrige el factor de escala.

$$\begin{bmatrix} A_{x1} \\ A_{y1} \\ A_{z1} \end{bmatrix} = [A_m]_{3 \times 3} \begin{bmatrix} 1/A_{SC_x} & 0 & 0 \\ 0 & 1/A_{SC_y} & 0 \\ 0 & 0 & 1/A_{SC_z} \end{bmatrix} \cdot \begin{bmatrix} A_x - A_{OS_x} \\ A_y - A_{OS_y} \\ A_z - A_{OS_z} \end{bmatrix}$$
$$= \begin{bmatrix} ACC_{11} & ACC_{12} & ACC_{13} \\ ACC_{21} & ACC_{22} & ACC_{23} \\ ACC_{31} & ACC_{32} & ACC_{33} \end{bmatrix} \cdot \begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix} + \begin{bmatrix} ACC_{10} \\ ACC_{20} \\ ACC_{30} \end{bmatrix}$$

En este caso aunque parece bastante compleja se simplifica mucho durante el proceso. Como explicaré a continuación, el procedimiento seguido permite simplificar las matrices de la siguiente manera.

$$\begin{bmatrix} A_{x1} & A_{y1} & A_{z1} \end{bmatrix} = \begin{bmatrix} A_x & A_y & A_z & 1 \end{bmatrix} \cdot \begin{bmatrix} ACC_{11} & ACC_{21} & ACC_{31} \\ ACC_{12} & ACC_{22} & ACC_{32} \\ ACC_{13} & ACC_{23} & ACC_{33} \\ ACC_{10} & ACC_{20} & ACC_{30} \end{bmatrix}$$

Donde la matriz de la derecha es la de calibración, que se obtiene en los apartados siguientes.

Procedimiento de recogida de valores

Por suerte, el modo de solucionar los tres problemas al mismo



tiempo es muy sencillo. Solamente hay que poner el acelerómetro en 6 posiciones de forma que cada eje alcance su valor máximo y los restantes valgan cero (en dirección a la gravedad). Para ello hay que seguir la siguiente tabla:

Stationary position	Accelerometer (signed integer)			Magnetic sensor (signed integer)		
	A_x	A_y	A_z	M_x	M_y	M_z
Z_b down	0	0	+1g	+ or -	+ or -	+
Z_b up	0	0	-1g	+ or -	+ or -	-
Y_b down	0	+1g	0	+ or -	+	+ or -
Y_b up	0	-1g	0	+ or -	-	+ or -
X_b down	+1g	0	0	+	+ or -	+ or -
X_b up	-1g	0	0	-	+ or -	+ or -

Estos vectores (a_x, a_y, a_z) se almacenan como w_1, w_2, w_3, w_4, w_5 , y w_6 . Se crean unos vectores Y para cada una de esas posiciones, los cuales contienen el valor calibrado que debería tener esa posición.

Así pues, en el caso de Z_b up se tiene un vector w_1 cuyo valor debería ser igual a $[0 \ 0 \ 1]$. Hay que asociar los valores del vector w_1 con los de Y_1 para compensar todos los errores y normalizar los datos, haciendo que estén comprendidos entre $+/-1$.

$$Y_1 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}_{n1 \times 3}$$
$$w_1 = \begin{bmatrix} A_{xP1} & A_{yP1} & A_{zP1} & 1 \end{bmatrix}_{n1 \times 4}$$

Para hacer esta imposición con todos los vectores a la vez se



incluyen en una matriz Y y una W de la siguiente forma:

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \\ Y_5 \\ Y_6 \end{bmatrix}_{nx3}$$

$$W = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \end{bmatrix}_{nx4}$$

Al final lo que de pretende hacer es una ecuación bien sencilla:

$Y = w * X$, osea $X = Y / w$ donde X es la matriz de calibración.

Para lograrlo se hace un ajuste por mínimos cuadrados. Quedaría así:

$$X = [w^T \cdot w]^{-1} \cdot w^T \cdot Y$$

Función de obtención de la matriz de calibración

```
function [ACC]=getAccCalibMatrix(w1, w2, w3, w4, w5, w6)
```

Contiene exactamente el procedimiento descrito en el apartado anterior pero implementado en forma de código. A la matriz X para

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	---------------------------

evitar confusiones con los ejes se le ha llamado ACC.

Función para calibrar y normalizar los datos recibidos

```
function [Ax1,Ay1,Az1]=normalize_acc(Ax, Ay, Az, ACC)
```

Simplemente recoge los datos crudos del acelerómetro y los multiplica por la matriz de calibración. De este modo a la salida obtenemos los valores calibrados y normalizados entre -1 y 1.

B- Calibración del magnetómetro:

Explicación teórica.

La calibración del magnetómetro es algo más compleja que la anterior. Se basa en principios bastante similares, pero debe hacerse paso a paso.

Para lograrlo hay que recoger datos de tres giros completos del sensor en cada eje. El resultado idealmente sería una matriz de puntos de forma esférica y centrada en los ejes. No obstante, en la práctica puede encontrarse distorsionada, obteniendo forma elíptica y descentrada de los ejes. Esta forma se respondería con la siguiente ecuación:

$$\frac{(x - x_0)^2}{a^2} + \frac{(y - y_0)^2}{b^2} + \frac{(z - z_0)^2}{c^2} + \frac{(x - x_0)(y - y_0)}{d^2} + \frac{(x - x_0)(z - z_0)}{e^2} + \frac{(y - y_0)(z - z_0)}{f^2} = R^2$$

No obstante, como se explicará más adelante, ciertos efectos se

 <p>Escuela de Ingeniería y Arquitectura Universidad Zaragoza</p>	<p>Sistema de monitorización de la posición de cámaras en rodajes 3D</p>	<p>Memoria 14/11/2011</p>
--	--	-------------------------------

pueden despreciar de modo que quedaría una ecuación más simplificada.

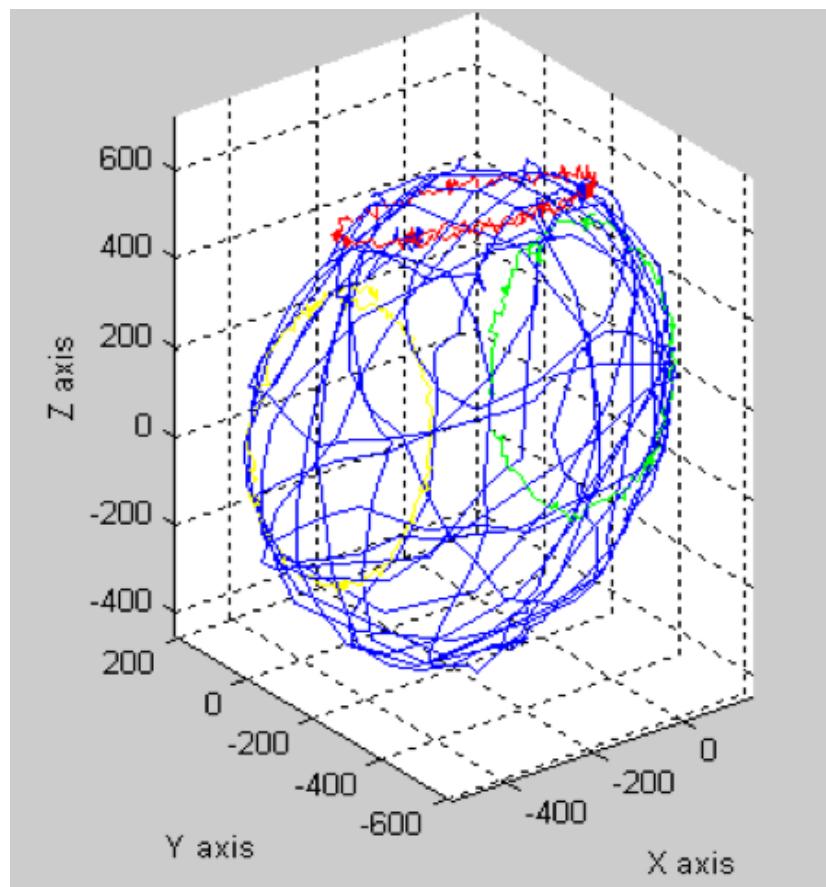
$$\frac{(x - x_0)^2}{a^2} + \frac{(y - y_0)^2}{b^2} + \frac{(z - z_0)^2}{c^2} = R^2$$

Conocidos todos los valores de la ecuación se pueden implementar unas matrices de calibración que corrijan el factor de escala y el offset.

A continuación se explican los dos efectos más importantes que pueden aparecer en un sensor de este tipo. Para dichos efectos se han conservado sus nombres en inglés, puesto que no se ha encontrado ninguna traducción al castellano.

Efectos soft-iron.

Son inducidos por materiales que producen distorsiones o influencias en el campo magnético. Se pueden distinguir gráficamente porque convierten la esfera en un elipsoide típicamente inclinado respecto a los ejes, como se muestra a continuación.



La imagen escogida no ha sido capturada durante los experimentos, puesto que la esfera de datos perteneciente al sensor utilizado era prácticamente perfecta. Esto ha permitido obviar estos efectos, como ya predecía el datasheet. De hecho el fabricante daba por hecho que los efectos serían nulos y no incluía mayor información al respecto.

Por lo que se ha podido averiguar, la compensación de este efecto es muy compleja, en ocasiones incluso imposible. La manera de conseguirlo sería multiplicando cada valor del elipsoide por una

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

constante, de tal forma que el resultado fuera una esfera perfecta.

Desgraciadamente estos efectos no suelen ser constantes y por tanto esta técnica no siempre es válida.

Efectos hard-iron [3]

Son de lejos los más importantes en los magnetómetros puesto que su aparición es muy habitual. Este tipo de distorsión se suele producir por materiales que añaden un campo magnético constante al propio campo magnético de la tierra, y por tanto suman un valor constante (offset) a cada uno de los ejes del magnetómetro.

Si por ejemplo el sensor se encontrara en la PCB de un teléfono móvil o un portátil, el imán permanente de los altavoces podría producir esta distorsión.

En caso de representar este efecto visualmente se apreciaría un esfera perfecta pero cuyo centro no coincidiera con el eje de coordenadas.

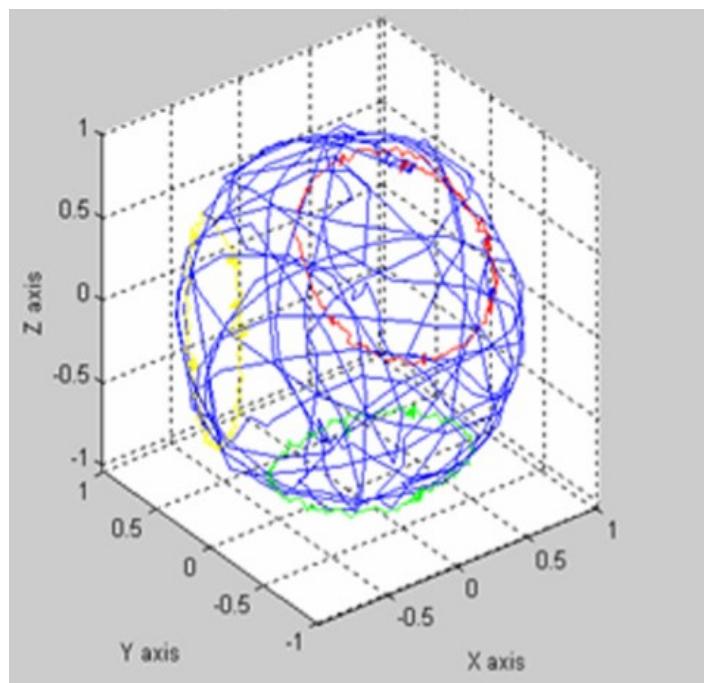
Procedimiento de calibración.

Como se ha explicado anteriormente, el primer paso para realizar la calibración es obtener datos del sensor. Para ello se deben dar un mínimo de tres giros completos alrededor de los tres ejes del magnetómetro.

Aprovechando que la calibración no es un proceso en tiempo real,



se ha aplicado un complejo y pesado filtro digital para eliminar el ruido de las medidas. Concretamente se trata del filtro “smooth” incluido en Matlab. Es un algoritmo de media móvil ponderada. El factor de ponderación puede ser modificado por el usuario hasta encontrar el compromiso perfecto en el que se filtre sin falsear las medidas.



Tras hacer las tres rotaciones completas en 2D y recolectar las medidas crudas del sensor magnético, se puede realizar un desarrollo matemático que convierta la ecuación de la esfera en la siguiente ecuación matricial:



$$\frac{(x - x_0)^2}{a^2} + \frac{(y - y_0)^2}{b^2} + \frac{(z - z_0)^2}{c^2} = R^2$$



$$x^2 = \begin{bmatrix} x & y & z & -y^2 & -z^2 & 1 \end{bmatrix} \cdot \begin{bmatrix} 2x_0 \\ \frac{a^2}{b^2} 2y_0 \\ \frac{a^2}{c^2} 2z_0 \\ \frac{a^2}{b^2} \\ \frac{a^2}{c^2} \\ a^2 R^2 - x_0^2 - \frac{a^2}{b^2} y_0^2 - \frac{a^2}{c^2} z_0^2 \end{bmatrix}$$

Donde :

-x, y, z son valores 2D de los distintos ejes del magnetómetro (M_x, M_y, M_z);
-x₀, y₀, z₀ son las constantes de offset (que se averiguan para la matriz M_OS)

-a, b, c son las longitudes de los semiejes de la esfera (necesarias para la matriz M_SC).

-R es una constante de la fuerza del campo magnético terrestre.

La ecuación anterior sería en el caso de una sola medida. Para tratar los datos en forma matricial, el aspecto sería el siguiente:

$$w_{nx1} = [H]_{nx6} \cdot x_{6x1}$$

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

Donde X sería la matriz de incógnitas, mientras que las otras se construyen con los datos adquiridos.

La manera de hallar los valores de la matriz X es similar a la empleada en la calibración del acelerómetro, por el método de mínimos cuadrados. De este modo la matriz H está bien condicionada y semidefinida positiva (tiene el rango completo y se puede invertir):

$$X = [H^T H]^{-1} H^T \cdot w$$

En la práctica, el paso a seguir para obtener la matriz de offset M_OS consiste en despejar x0, y0 y z0. Del primer valor de la matriz X (2*x0) se deduce x0. Con el segundo y el cuarto sale y0. Con el tercero y el quinto se obtiene z0.

$$M_{OS_x} = x_0 = X(1)/2$$

$$M_{OS_y} = y_0 = X(2)/(2 \cdot X(4))$$

$$M_{OS_z} = z_0 = X(3)/(2 \cdot X(5))$$

Para obtener la matriz para reescalar los valores, M_SC, hay que obtener los valores A, B y C correspondientes a los denominadores de la ecuación de la esfera. En primer lugar hay que despejar R^2 e introducirla en dichos denominadores. Luego se despejan A, B y C igual que en caso de M_OS:



$$A = a^2 R^2 = X(6) + x_0^2 + X(4) \cdot y_0^2 + X(5) \cdot z_0^2$$

$$B = A / X(4)$$

$$C = A / X(5)$$

Si se hace ya la resta de las medidas crudas y el offset así:

$$xx = M_x - M_OS_x$$

$$yy = M_y - M_OS_y$$

$$zz = M_z - M_OS_z$$

La ecuación de la esfera respondería a la siguiente forma:

$$\frac{xx^2}{A} + \frac{yy^2}{B} + \frac{zz^2}{C} = 1$$

Por tanto para aplicar el factor de escala a cada componente tenemos que tener en cuenta que la ecuación de una esfera ideal es

$x^2 + y^2 + z^2 = 1$ luego los valores de factor de escala se corresponden con la raíz cuadrada:

$$M_SC_x = \sqrt{A}$$

$$M_SC_y = \sqrt{B}$$

$$M_SC_z = \sqrt{C}$$

De este modo ya se han obtenido las matrices necesarias para la calibración.

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

Función de obtención de la matriz de calibración

```
function [ X, M_OS, M_SC ]=getMagCalibMatrix(M)
```

Exactamente igual que ocurría en el caso del acelerómetro, esta función calcula la matriz de calibración para el magnetómetro siguiendo los pasos explicados en el apartado anterior.

No obstante, en vez de poder dar una matriz compacta que solucione todos los problemas, la función devuelve la matriz de corrección de offset (M_OS) y la del factor de escala (M_SC).

Función para calibrar y normalizar los datos recibidos

```
function [ Mx1, My1, Mz1 ]=normalize_mag(Mx, My, Mz, M_OS, M_SC)
```

Sencillamente coge las medidas crudas del magnetómetro y les resta los valores de offset. Después multiplica esa nueva matriz sin efectos *hard-iron* por M_SC que corrige el factor de escala. De este modo a la salida obtenemos los valores calibrados y normalizados entre -1 y 1.

2.3.6 Función de cálculo de inclinación.

A partir de los datos calibrados hay que ejecutar una función que permita el cálculo de inclinación en el eje X (pitch) y en el Y (roll) en radianes. Por tanto los datos de entrada de esta función serán Ax1, Ay1 calibrados y normalizados en la función "normalize_acc".

Una vez obtenidos, la función a aplicar es la siguiente:

```
function [ pitch, roll ]=position(Ax1, Ay1)
```

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

La función aplica unas sencillas funciones trigonométricas que convierten los datos a radianes. Son las siguientes:

```
pitch = asin(-Ax1)
roll = asin(Ay1/cos(pitch))
```

Además hay que tener cuidado con el coseno de +-pi/2 que puede dar problemas, porque tendríamos arcsen (Ay1 / 0).

Como se puede observar devuelve "pitch" y "roll" que son la inclinación en eje X y en Y respectivamente. Con estos dos datos se puede simular un nivel de burbuja, como el utilizado en las obras.

Además de darnos la inclinación, también se utilizan estos dos datos para la siguiente función.

2.3.7 Función de cálculo de cabeceo.

```
f [heading,check_mag]=heading(Mx1, My1, Mz1, pitch, roll)
```

El último y mayor objetivo de todas estas funciones es obtener una manera fiable y precisa el cabeceo del sensor. Esto quiere decir que se sabrá en todo momento la posición del mismo respecto del norte magnético, incluso si aparecen inclinaciones que pudieran falsear la medida.

En primer lugar, los parámetro de inclinación *pitch* y *roll* permiten realizar esa compensación. La expresión a aplicar es la siguiente:

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

$$M_{x2} = M_{x1} \cos \rho + M_{z1} \sin \rho$$

$$M_{y2} = M_{x1} \sin \gamma \sin \rho + M_{y1} \cos \gamma - M_{z1} \sin \gamma \cos \rho$$

$$M_{z2} = -M_{x1} \cos \gamma \sin \rho + M_{y1} \sin \gamma + M_{z1} \cos \gamma \cos \rho$$

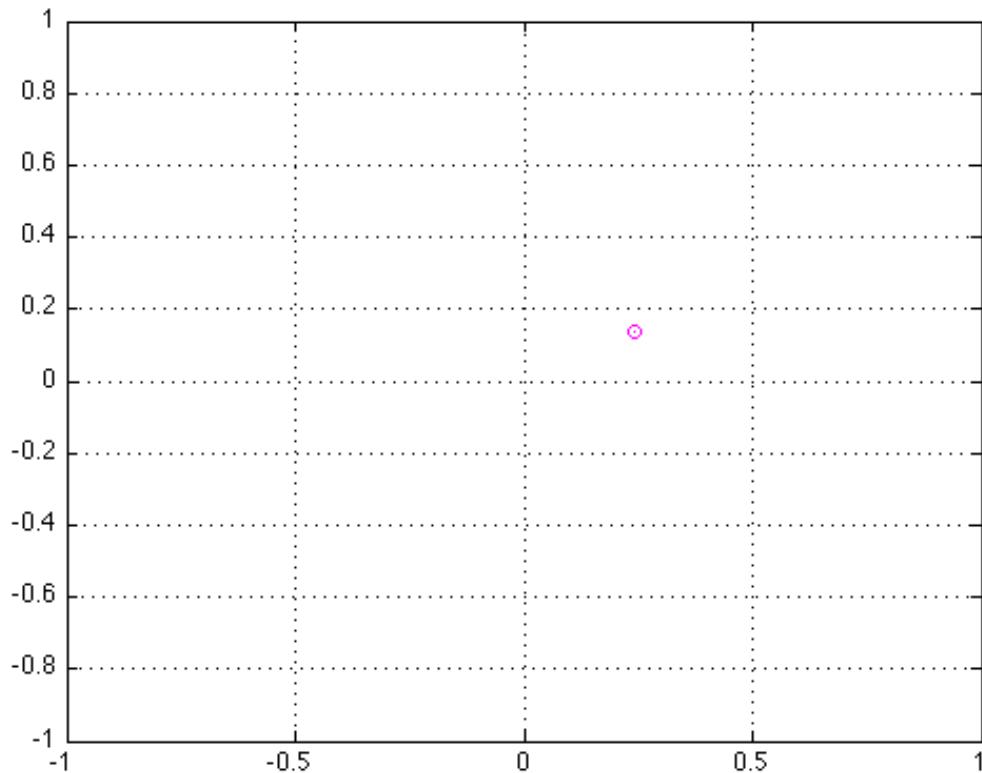
Siendo las M_{n1} las medidas del magnetómetro calibradas y las M_{n2} los mismos parámetros después de la compensación de inclinación.

El cálculo del cabeceo ya es trivial mediante el $\arctan(M_{y2} / M_{x2})$. No obstante hay que tener en cuenta los cambios de signo de los valores del magnetómetro de cara al resultado del \arctan . En algunos casos hay que sumar 180° para conseguir que la salida este comprendida entre 0° y 360° , así como añadir excepciones en 90° y 270° para sortear las asíntotas.

2.3.8 Representación gráfica en tiempo real.

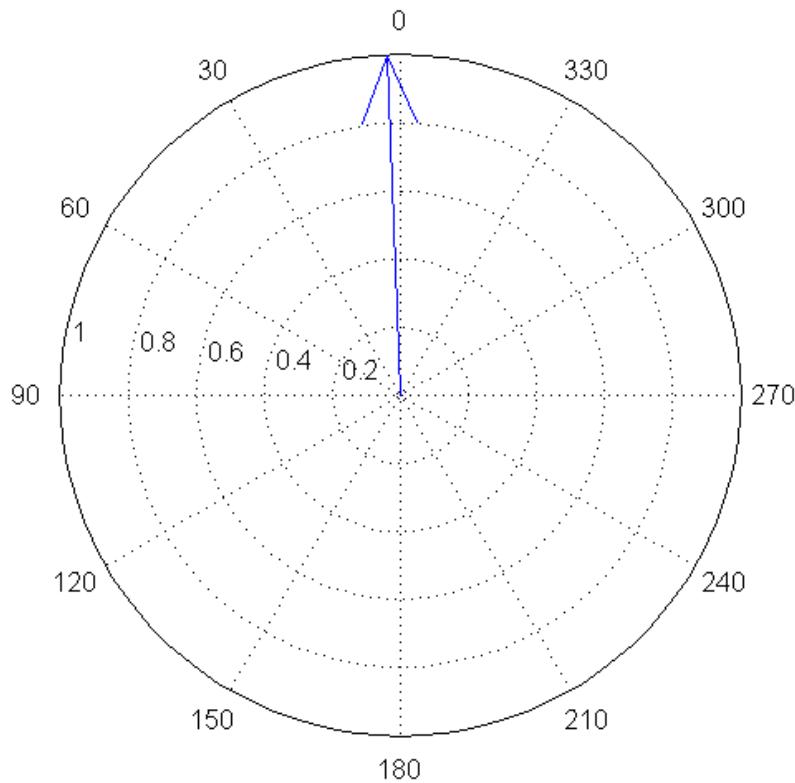
Para la representación en tiempo real se va a mostrar lo siguiente:

-Gráfica con la inclinación en los ejes X e Y. El aspecto es de un nivel de burbuja. Para ello se usa la función *plot* típica de Matlab. En la imagen los ejes están normalizados a la unidad, pero se puede ver también en grados.



-Simulación de la brújula digital. Es una representación gráfica del cabeceo. Como se puede observar ya se ha representado en grados para que parezca una brújula real.

La función utilizada también es originaria de Matlab y se llama *compass*.



Filtro previo:

Actualmente se está trabajando para filtrar el ruido en los valores de pitch, roll y heading. Al tener que realizarse en tiempo real, es mejor para el procesador el filtrar estos 3 resultados en vez de las 6 medidas brutas recogidas del sensor.

Probablemente el filtro será de tipo *smooth*, que consiste en una media móvil bastante eficiente. No obstante la dificultad reside en que no haya mucho retraso entre el movimiento real y el de las gráficas.

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

3.CONCLUSIONES

3.1 Resultados finales

En el apartado de hardware se ha logrado una PCB de sensores de un tamaño muy reducido, que resulta despreciable respecto a la placa de Arduino. Se han atornillado ambas placas y se ha dotado al prototipo de cierta estructura para ganar en estabilidad.

En cuanto al software, se han cumplido de forma muy aceptable con los requisitos previos.

-Se ha conseguido adquirir sin ningún problema los datos del sensor, tratarlos y representarlos gráficamente en tiempo real, a pesar de las limitaciones de velocidad de la conexión serie.

-La calibración ha sido un éxito, consiguiendo eliminar los problemas presentes en los sensores. Especialmente importante ha sido la compensación del efecto "hard-iron" en el magnetómetro.

-Se han creado scripts de calibración para que sea fácil y rápido su uso en caso de utilizar otro sensor. Para cada nuevo sensor hay que aplicar estas funciones y de este modo se asegura que las medidas tomadas son fidedignas.

-El filtrado y la calibración han permitido reducir el error en las medidas. Actualmente se están realizando las pruebas que determinarán si se puede lograr la representación el tiempo real con un error menor de 1º. La incursión del filtro de Matlab ralentiza

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	---------------------------

considerablemente la representación gráfica.

-El sistema es totalmente multiplataforma ya que se basa en Matlab (disponible en Windows, Linux y Mac OS). La única variación en el código consiste en que hay que modificar el puerto serie al que se conectan las placas Arduino, ya que en cada ordenador es diferente.

3.2 Futuras mejoras

Se ha repetido bastante a lo largo de la memoria que el sistema está preparado para futuras ampliaciones para llegar a lograr un sistema completo y comercial. A continuación se habla de ellas.

El siguiente paso lógico es añadir un giroscopio que mide la velocidad de giro mientras este se produce. No son demasiado precisos para los movimientos suaves como los producidos en el trípode, pero en caso de situar las cámaras en una grúa serían ideales. Además un movimiento brusco falsearía la medida del acelerómetro, y este a su vez el cálculo de cabeceo. Con este sensor se podría pensar un algoritmo que permitiera sortear este problema.

La idea es trabajar con un giroscopio de tres ejes por I2C, que se integraría perfectamente en el sistema actual sin demasiadas modificaciones de software.

Otro posible añadido vendría de la mano de los railes utilizados para hacer “travelling”. La cámara se desplaza sobre ellos a velocidades

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

bastante elevadas en una trayectoria conocida. Se ha pensado en incluir un encoder relativo en una de las ruedas para saber cuanto se desplaza en el tiempo con una gran precisión. Sólo usando una entrada digital de Arduino ya se obtendría una medida excelente y muy útil.

3.3 Conclusión personal

Se puede afirmar sin ninguna duda que se han logrado los objetivos planteados y se ha dado un paso de gigante hacia la consecución de un producto terminado que unifique toda la tecnología desarrollada por la empresa 3+D.

Ha sido un camino largo en el que he aprendido mucho. Desde los fundamentos de la programación de alto nivel hasta el cine 3D, pasando por Matlab, Arduino o el funcionamiento de los sensores magnéticos.

Muchas veces he tenido que detenerme más de la cuenta en pequeños errores y también se ha dado el caso de toparme con una pared insalvable y tener que recular. Sin embargo no se me ocurre una forma mejor de aprender que los errores de cada uno, así como de las aportaciones de los compañeros, que siempre tienen algo que enseñar.

En definitiva ha sido una experiencia dura pero enriquecedora. Si volviera a empezar lo haría todo distinto, y más rápido, pero seguramente esto sea algo innato a la profesión del ingeniero.

 <p>Escuela de Ingeniería y Arquitectura Universidad Zaragoza</p>	<p>Sistema de monitorización de la posición de cámaras en rodajes 3D</p>	<p>Memoria 14/11/2011</p>
--	--	-------------------------------

Agradecimientos:

Quisiera agradecer aquí la paciencia de mis padres, a los que día tras día les tuve que decir que no me daba tiempo a comer en casa, y siempre a última hora.

Por supuesto debo dar también mil gracias a Álvaro y Patxi, compañeros de 3+D, que siempre estaban ahí para ayudarme con los problemas en las partes más peliagudas y sobre todo en los comienzos con Maltab.

Me gustaría mandar un saludo a mi director Jesús Ponce y a Antonio Bono que se ofreció amablemente a ser ponente del proyecto. También a los compañeros de laboratorio, esos que compartimos tantos dolores de cabeza durante estos meses.

Por último, quiero dedicar esta línea de agradecimientos a Tomás Pollán. Por ser un docente que inspiró tanto a tantos. Los que te conocieron siempre hablarán de ti con buenas palabras.

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

4. REFERENCIAS Y GLOSARIO

4.1 REFERENCIAS.

-Documentación

En esta sección se incluyen enlaces que han servido para documentar sobre diversas cuestiones principalmente teóricas. También se contemplan temas prácticos pero no a nivel de datasheet o código de programación.

3D:

[1] Teoría del 3D:

<http://www.3d-forums.com/stereoscopic-parallax-t4.html>

[2] Historia del 3D:

<http://www.kinepolis.com/es/index.cfm?PageID=108230>

-Ambos:

Libro "3D Movie Making: Stereoscopic digital cinema from script to screen" de *Bernard Mendiburu*. Elsevier 2009.

Sensores magnéticos:

[3] Compensación de efectos hard-iron y soft-iron:

[http://www.sensorsmag.com/sensors/motion-velocity-
displacement/compensating-tilt-hard-iron-and-soft-iron-effects-](http://www.sensorsmag.com/sensors/motion-velocity-displacement/compensating-tilt-hard-iron-and-soft-iron-effects-)

 <p>Escuela de Ingeniería y Arquitectura Universidad Zaragoza</p>	<p>Sistema de monitorización de la posición de cámaras en rodajes 3D</p>	<p>Memoria 14/11/2011</p>
--	--	-------------------------------

6475

[4] Sobre el campo magnético terrestre:

http://www.telefonica.net/web2/jgarciaf/cambio_climatico/Campo_magntico/cmagnetico.htm

http://en.wikipedia.org/wiki/Earth's_magnetic_field

Wikipedia:

[5] Información sobre el filtro de media móvil:

http://en.wikipedia.org/wiki/Moving_average

[6] Croma: <http://es.wikipedia.org/wiki/Croma>

[7] Esteroscopía:

<http://es.wikipedia.org/wiki/Estereoscop%C3%A3Da>

Arduino:

[8] Todo sobre Arduino desde lo divulgativo hasta lo más técnico:

<http://www.arduino.cc>

-Código de apoyo

[9] Librería I2C de Arduino: <http://arduino.cc/en/Reference/Wire>

[10] Librería Matlab (arduino.m):

<http://www.mathworks.es/academia/arduino-software/arduino-matlab.html>

-Datasheets

	<p>Escuela de Ingeniería y Arquitectura Universidad Zaragoza</p>	<p>Sistema de monitorización de la posición de cámaras en rodajes 3D</p>	<p>Memoria</p>
			14/11/2011

[11] Datasheet LSM303DLH:

[http://www.pololu.com/file/download/LSM303DLH.pdf?
file_id=0J433](http://www.pololu.com/file/download/LSM303DLH.pdf?file_id=0J433)

[12] Nota para aplicación del LSM303DLH como brújula:

[http://www.pololu.com/file/download/LSM303DLH-compass-app-
note.pdf?file_id=0J434](http://www.pololu.com/file/download/LSM303DLH-compass-app-note.pdf?file_id=0J434)

[13] Datasheet Atmel Atmega328P:

[http://www.atmel.com/dyn/resources/prod_documents/doc8161.p
df](http://www.atmel.com/dyn/resources/prod_documents/doc8161.pdf)

[14] Datasheet Micrel mic5205:

http://www.micrel.com/_PDF/mic5205.pdf

-Compra de material

- Farnell: <http://es.farnell.com/>
- Pololu: <http://www.pololu.com/>
- Sparkfun: <http://www.sparkfun.com/>

-Otras

[16] Noticia relacionada con Arduino:

[http://www.elpais.com/articulo/Pantallas/Internet/cosas/crece/el
pepurty/20110713elpepirty_3/Tes](http://www.elpais.com/articulo/Pantallas/Internet/cosas/crece/el
pepurty/20110713elpepirty_3/Tes)

4.2 SOFTWARE Y GESTIÓN

4.2.1 Software utilizado:

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	---------------------------

Programas de cálculo:

-Matlab R2010b para el tratamiento de datos de sensores, gráfica y analíticamente. Además contiene la interfaz de comunicación con Arduino.

-Microsoft Excel 2007 para cálculo de presupuestos.

Entornos de desarrollo (programación):

-Arduino IDE para la programación de la placa Arduino.

-Codeblocks para compilar código cpp.

Programas de diseño:

-Eagle 2011 (CadSoft) para el diseño de la PCB.

4.2.2 Comunicación interna y sistemas colaborativos en la empresa.

Mención aparte requiere la gestión que se ha hecho de la información del proyecto gracias a los nuevos medios informáticos.

Partiendo de la base de que la empresa contratante es una startup sin apenas infraestructura física y con muy pocos trabajadores, no se disponía de ninguna herramienta ERP¹⁶. Esto está totalmente justificado puesto que las dimensiones de la empresa no son lo suficientemente grandes como para amortizar semejante inversión. Por otra parte hay que aclarar que los mismos empleados realizan varios mini-proyectos en paralelo a este (y entrelazados con él) para completar la funcionalidad del sistema.

En vista de todo esto, se optó por el uso de la plataforma online

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

que nos ofrece Google para trabajo en grupo. Aprovechando la aparición de Google+ este verano (<https://plus.google.com/>) se decidió exprimir toda su funcionalidad.

Por un lado Google+ es una red social, muy similar a Facebook, pero su enorme facilidad para la gestión de la privacidad hace que sea compatible tanto para el ocio como para el trabajo. En este sentido se ha usado para comunicación síncrona (chat y videoconferencia múltiple, pizarra digital compartida...) y asíncrona (conversaciones privadas multipersonales, compartición de links y archivos, etc). Esto se debe a su integración con "Google Docs", otra herramienta colaborativa de google que ya permitía la edición de documentos Office en tiempo real por varios usuarios a la vez y la compartición de archivos "on the cloud". Precisamente es el concepto de cloud computing¹⁷ el que hay que destacar como base de todo este planteamiento.

Una aplicación tan simple (y gratuita) como esta ha mejorado enormemente la productividad del grupo, puesto que es multidisciplinar y trabaja en distintos proyectos. Además, a pesar de trabajar muchas veces a distancia, aumentaba la cohesión y la sensación de respaldo, que a su vez influye en la motivación personal de cada uno.

4.3 GLOSARIO

1. Monitorización: *Por definición, monitorización es el proceso por medio*

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

del cual, nos aseguramos que nuestro proceder está encaminado adecuadamente y eficazmente hacia un resultado final, evitando las posibles desviaciones que pudieran presentarse. En el caso que nos ocupa se refiere a poder seguir en tiempo real el movimiento de la cámara de video mediante medidas de sensores, y además guardar esos datos de aceleraciones, ángulos de giro, etc para tenerlos en cuenta en la posterior edición de la grabación.

2. Rig: *La plataforma gracias a la cual se pueden anclar las cámaras de vídeo al trípode donde se sustentan. Suele ser de metal, y su calidad es determinante a la hora de rodar en 3D puesto que se usan dos cámaras perfectamente simétricas. Si por ejemplo se produjeran oscilaciones en una cámara pero en la otra fueran más leves, el resultado sería un gran deterioro de calidad en el efecto tridimensional.*



3. Croma: *[6] Es una técnica audiovisual utilizada ampliamente tanto en cine y televisión como en fotografía, que consiste en la sustitución de un fondo por otro mediante un equipo especializado o un ordenador. Esto se hace debido a que es demasiado costoso e inviable el recorte del fondo o personajes para completar cuadro a cuadro mediante rotoscopia.*



Para realizar esta operación con facilidad y buen resultado, se utiliza el Chroma key o "clave de color": Básicamente consiste en un fondo de color sólido y uniforme y el objeto que se desea recortar o cambiar de fondo, dejando lo demás para tarea del ordenador. Para realizar la operación correctamente se debe escoger un fondo y después un objeto o persona para colocar en una situación posterior a ese hipotético fondo.

 <p>Escuela de Ingeniería y Arquitectura Universidad Zaragoza</p>	<p>Sistema de monitorización de la posición de cámaras en rodajes 3D</p>	<p>Memoria 14/11/2011</p>
--	--	-------------------------------

Este color debe ser alguno de los primarios (rojo, verde o azul). Habitualmente se usa verde o azul y no rojo, ya que es el componente de color más importante de la piel humana. Aunque el color más habitual del fondo o ciclorama era el azul, el verde se ha impuesto al ser la ropa azul mucho más común que la verde.

(fuente Wikipedia)

4. Posproducción: *La posproducción es la manipulación de material audiovisual, normalmente posterior al rodaje. Se suele llamar posproducción a la aplicación de filtros, iluminación artificial, efectos especiales, animación, etc.*

5. Fiducial: *Se trata de la referencia visual que es capaz de reconocer un computador a través de la visión artificial. Dicha referencia puede ser desde una figura en papel hasta el rostro humano. El principal requisito es que la referencia disponga de una serie de marcas de tamaño y separación conocidos para permitir al computador identificar el objeto en cualquier situación.*

Por ejemplo en los códigos BIDI como el de la foto se reconoce el objeto gracias a los cuadrados de las esquinas.



6. Escenografía renderizada: *Consiste en cubrir todo o parte del escenario con croma (una tela verde), de tal manera que más tarde se pueda añadir el escenario deseado con posproducción digital. Es una técnica muy utilizada hoy en*

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

día en Hollywood (por ejemplo las nuevas películas de "Star Wars" están grabadas íntegramente en croma). También se pueden añadir personajes digitalmente que parecen interactuar con humanos.

La renderización es el paso que hay que dar para transformar los diseños tridimensionales (vectores y polígonos) a imágenes bidimensionales, que son realmente las que aparecen en la pantalla (pixeles, anchura x altura). Es un proceso que realizan computadores especialmente preparados puesto que consume enormes recursos y es muy difícil y caro lograr hacerlo en tiempo real.

7. Monitores de campo: *Son las pantallas que reproducen lo que está viendo la cámara en tiempo real durante el rodaje. El equipo de dirección y fotografía la usan para saber realmente como se están grabando las cosas e intervenir cuando sea necesario. Si la película es en 3D y además grabada con croma (como en el caso de Avatar) nos encontramos con un gran problema. Por una parte hay que proyectar en tiempo real en el monitor de campo la escenografía, aunque sea muy simplificada, para saber si los personajes interactúan bien con el entorno que es invisible para ellos. Además si son cámaras de 3D, no podemos olvidar que toda la configuración cambia si son desplazadas, y habrá que modificar de igual manera la escenografía proyectada en el monitor para saber si el efecto tridimensional es correcto o no (corregir esto a posteriori es costoso y en ocasiones imposible). De nuevo es imperativo conocer los movimientos de la cámara en tiempo real.*



8. Estereoscopía: [7] *Es cualquier técnica capaz de recoger información*

 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	-----------------------

visual tridimensional y/o crear la ilusión de profundidad en una imagen. La ilusión de la profundidad en una fotografía, película, u otra imagen bidimensional se crea presentando una imagen ligeramente diferente para cada ojo, como ocurre en nuestra forma habitual de ver. Muchas pantallas 3D usan este método para transmitir imágenes. Fue inventado por Sir Charles Wheatstone en 1840. (fuente Wikipedia).

9. Mainstream (corriente principal): *Se emplea al hablar de arte en general y de música moderna en particular, para designar los trabajos que cuentan con grandes medios para su producción y comercialización, y que llegan con facilidad al gran público. Un buen ejemplo de mainstream podría ser la cultura pop, producida principalmente para su comercialización y en muchos casos con el objetivo de obtener la mayor cantidad de ventas y beneficios económicos. En el caso del cine, el término se refiere básicamente a las super producciones de Hollywood.*

10. Tilt, roll y heading (inclinación, giro y cabeceo): *Se refieren a la rotación de acuerdo a los ejes cartesianos según la unificación propuesta por el sector aeronáutico. Durante todo el proyecto los estamos nombrando en inglés porque así aparecen en toda la bibliografía y son inequívocos.*

11. Wiimote: *Es el mando principal de la consola Wii de Nintendo. Destaca por ser capaz de captar movimientos con bastante precisión basándose en el uso de acelerómetros y giroscopios, así como de un sensor de infrarrojos que se comunica con una barra de sensores similares que se pone sobre la televisión.*

12. Kinect: *Es un accesorio para la consola "X-Box 360" de Microsoft consistente en una cámara que integra un potente software de visión por computador (además en código abierto para quien quiera darle un uso distinto del original). Su utilidad principal es el reconocimiento*



 Escuela de Ingeniería y Arquitectura Universidad Zaragoza	Sistema de monitorización de la posición de cámaras en rodajes 3D	Memoria 14/11/2011
--	--	---------------------------

de movimientos de los jugadores, permitiéndoles manejar los juegos sin mando.



13. Hardware libre: *Se llama hardware libre a los dispositivos de hardware cuyas especificaciones y diagramas esquemáticos son de acceso público, ya sea bajo algún tipo de pago o de forma gratuita. La filosofía del software libre (las ideas sobre la libertad del conocimiento) es aplicable a la del hardware libre. Se debe recordar en todo momento que libre no es sinónimo de gratis.*

Las implicaciones son muchas. Algunos ejemplos son la mayor supervisión de los contenidos gracias a la comunidad, aumentando su fiabilidad o el ahorro de tiempo y costes por no tener que repetir trabajo ya realizado por terceros.

14. Wiring: *Es una plataforma de código abierto de prototipado. Se compone de un entorno de desarrollo integrado (IDE), una placa de prototipado y la documentación creada por la comunidad.*

15. Processing: *Es un lenguaje de programación de código abierto para personas que quieren crear imágenes, animaciones e interacciones. Especialmente orientado a aprendizaje y prototipos.*

16. ERP (enterprise resource planning): *Son sistemas informáticos para la planificación de recursos empresariales. Son capaces de aglutinar toda la gestión de la infinidad de departamentos que componen una compañía en un software unificado.*

17. Cloud computing (computación en la nube): *Es un paradigma que permite ofrecer servicios de computación a través de internet. En nuestro caso*

 <p>Escuela de Ingeniería y Arquitectura Universidad Zaragoza</p>	<p>Sistema de monitorización de la posición de cámaras en rodajes 3D</p>	<p>Memoria</p>
		<p>14/11/2011</p>

significa que podemos gestionar la información y documentación de los distintos proyectos de la empresa desde cualquier terminal, ya que todo está en internet. Además aprovechamos las mismas herramientas para establecer una comunicación síncrona y asíncrona entre el equipo.

18. Sketch: *En la comunidad Arduino se le llama sketch al programa que ejecuta el microcontrolador. El código se escribe desde el entorno de desarrollo facilitado por la comunidad. Dicho entorno es muy sencillo e intuitivo, pero no ofrece demasiadas opciones para el debug. En algunos casos puede ser preferible escribir código en lenguaje C, utilizar una IDE de mayor calidad e intrucir despues dicho código como un "include" en el sketch.*