

UNIVERSIDAD POLITÉCNICA DE VALENCIA  
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA  
PROYECTO FINAL DE CARRERA

---

# Desarrollo de un sistema de Realidad Aumentada en dispositivos móviles

---



UNIVERSIDAD  
POLITECNICA  
DE VALENCIA

Escuela Técnica Superior de Ingeniería Informática



PROYECTO REALIZADO POR:  
**Carlos Alcarria Izquierdo**

DIRIGIDO POR:  
**Dra. M<sup>a</sup> Carmen Juan Lizandra**

VALENCIA, 2010



# Contenidos

<b>CAPÍTULO 1</b>	<b>MOTIVACIÓN Y OBJETIVOS.....</b>	<b>1</b>
1.1	INTRODUCCIÓN.....	1
1.2	OBJETIVOS.....	2
1.3	ESTRUCTURA DEL PROYECTO.....	3
<b>CAPÍTULO 2</b>	<b>ESTADO DEL ARTE DE LA REALIDAD AUMENTADA EN DISPOSITIVOS MÓVILES.....</b>	<b>5</b>
2.1	REALIDAD AUMENTADA.....	5
2.2	REALIDAD AUMENTADA MÓVIL.....	7
2.3	ANTECEDENTES.....	10
2.4	DISPOSITIVOS PORTÁTILES.....	15
2.4.1	SITUACIÓN DEL MERCADO.....	15
2.4.2	TIPOS DE DISPOSITIVOS.....	17
2.5	CAMPOS DE APLICACIÓN.....	23
2.6	EJEMPLOS DE APLICACIONES.....	28
2.7	SISTEMAS SIN MARCADORES.....	43
2.8	LIBRERÍAS DE DESARROLLO.....	44
<b>CAPÍTULO 3</b>	<b>IMPLEMENTACIÓN DEL PROYECTO.....</b>	<b>50</b>
3.1	INTRODUCCIÓN.....	50
3.2	DESCRIPCIÓN DEL ANTERIOR PROYECTO.....	51
3.3	DISEÑO DE LA APLICACIÓN.....	56
3.3.1	DISPOSITIVO MÓVIL: IPHONE.....	56
3.3.2	HERRAMIENTAS SOFTWARE.....	58
3.3.2.1	IPHONE SDK.....	58
3.3.2.2	ARTOOLKITPLUS.....	61
3.3.2.3	BLENDER.....	64
3.3.2.4	OPENGL ES.....	66
3.3.3	ELEMENTOS VIRTUALES.....	67
3.3.4	FUNCIONAMIENTO DE ARTOOLKITPLUS.....	69
3.3.4.1	MARCADOR UTILIZADO.....	73
3.4	FUNCIONALIDADES.....	74
3.4.1	INICIAR ESCENA.....	75
3.4.2	AÑADIR/ELIMINAR OBJETOS.....	75
3.4.3	MOVIMIENTO.....	76
3.4.4	MODO COGER.....	77
3.4.5	SIMULAR ESPRAY.....	78
3.4.6	PARADA.....	79
3.4.7	AUMENTAR TAMAÑO.....	79
3.4.8	MOVIMIENTO MEJORADO.....	80
3.4.9	MODO COGER Y ELIMINAR.....	80
<b>CAPÍTULO 4</b>	<b>CONCLUSIONES Y TRABAJOS FUTUROS.....</b>	<b>81</b>
<b>CAPÍTULO 5</b>	<b>BIBLIOGRAFÍA.....</b>	<b>83</b>

# Capítulo 1

## Motivación y objetivos

### 1.1 Introducción

Uno de los objetivos principales de este proyecto es la implementación y validación de un sistema de Realidad Aumentada (en adelante RA) en un dispositivo móvil actual para la ayuda al tratamiento de la fobia a animales pequeños, en nuestro caso los insectos del orden de los blatodeos, o más comúnmente denominados, cucarachas. Asimismo también se pretende valorar la implementación de este tipo de sistemas en los nuevos dispositivos portátiles tales como los Smartphone, pues son los dispositivos que más se han extendido y los que más crecimiento han tenido en los últimos años. Otro de los objetivos del presente trabajo, es la realización de un estudio del estado del arte de la RA móvil, destacando principalmente los dominios y algunas de las aplicaciones que se han desarrollado en este sentido, las principales librerías actuales de desarrollo para estos sistemas y una breve reseña de los hitos científicos más relevantes que han posibilitado el desarrollo de la RA en dispositivos móviles.

Se ha optado por elegir el dispositivo iPhone, de la compañía estadounidense Apple Inc. Este Smartphone, es a fecha de la realización del proyecto, uno de los teléfonos móviles más versátiles y potentes que existen en el mercado. Nos brinda un entorno de programación flexible e intuitivo que nos va a permitir realizar el diseño del programa de una manera profesional y empleando herramientas ampliamente utilizadas en el mundo de la programación de dispositivos móviles.

La librería que se ha utilizado es ARtoolKitPlus (ARPLUS07), por sus virtudes probadas en otras aplicaciones creadas anteriormente. ARtoolKitPlus es una optimización de la librería ARtoolKit (ARTOOL99), pues se han redefinido algunas de sus funciones principales para adaptarlas a la potencia de cálculo de estos pequeños dispositivos. Para representar la información gráfica tal como son las cucarachas, se ha utilizado OpenGL ES, que es un conjunto reducido de funciones OpenGL optimizado. Para integrar todo ello y realizar la implementación, se ha utilizado el entorno de desarrollo propio de Apple, Xcode, junto con la SDK 3.1.3 para el desarrollo con el móvil iPhone.

La aplicación incluirá alguna de las funcionalidades disponibles en un proyecto anteriormente realizado, pero diseñado para un equipo de sobremesa. Estas funciones, serán las de dotar de movimiento a las cucarachas, permitir añadir o quitarlas de la escena de visión (la captura de vídeo de la cámara) y la posibilidad de simular el sonido de un espray para eliminarla, entre otras funciones. Todo ello se verá detalladamente en los capítulos correspondientes.

## 1.2 Objetivos

Dentro del campo de la RA, trabajaremos en este proyecto en un ámbito distinto como son las plataformas móviles. Uno de los objetivos principales será implementar y validar una aplicación basada en RA y que a su vez se haya desarrollado para un dispositivo móvil.

Definido cual será el ámbito donde nos moveremos, pasaremos a explicar los objetivos principales que han motivado este trabajo. Por un lado, contamos con una aplicación funcional que se encarga de servir de herramienta de ayuda para el tratamiento de fobias a las cucarachas. Se realizará una implementación que se ajuste lo máximo posible a la ya creada. Deberemos exigir unos criterios para que la aplicación pueda ser análoga a la que funciona sobre equipos de sobremesa, teniendo en cuenta las limitaciones de hardware de las que se dispondrá, ya que en cuanto a

hardware estamos más limitados, en software estaremos en prácticamente las mismas condiciones.

Los objetos 3D deberán ser realistas y realizar un movimiento convincente, que nos dé la sensación de realismo y además todo ello sin producir ningún retraso visual aparente.

Finalmente, otro de los objetivos será el estudio del arte de la RA en los dispositivos móviles. Para ello se realizarán consultas en las bases de datos de publicaciones científicas más importantes a fin de recabar información. Todo ello se presentará en un capítulo de este trabajo.

Estaremos pues, ante un proyecto que implica no sólo la búsqueda de información relevante en esta línea de investigación, sino también un reto que se deberá resolver de manera profesional y aplicando los conocimientos adquiridos a lo largo de la titulación, que es, de manera general, el objetivo principal de los proyectos final de carrera.

### 1.3 Estructura del proyecto

Veamos a continuación cómo está estructurado este documento. Se realizará un breve resumen de los contenidos que se han expuesto en el índice.

- El primer capítulo, *Introducción y Objetivos del proyecto*, se exponen cuáles van a ser los objetivos que han motivado la realización de este proyecto, así como también se hace una introducción sobre los contenidos que se han desarrollado.
- El segundo capítulo, llamado *Estado del arte de la RA móvil*, se realiza un estudio sobre la RA aplicada sobre los dispositivos móviles. Trataremos temas como los dominios de aplicación, librerías de desarrollo y sistemas desarrollados.
- El tercer capítulo, denominado *Implementación del proyecto*, es donde trataremos todo el proceso de desarrollo de la aplicación que se ha realizado en este proyecto. Haremos hincapié en las herramientas

## Capítulo 1: Motivación y objetivos

utilizadas, la metodología empleada y concluiremos con las funcionalidades que tiene la aplicación.

- El cuarto capítulo, *Resultados*, expondremos los resultados sobre el iPhone, viendo el rendimiento observado así como la validación de todas las funcionalidades.
- En el quinto capítulo, *Conclusiones y trabajos futuros*, comentaremos cuáles han sido las conclusiones obtenidas, además de exponer las posibles ampliaciones y los trabajos futuros.

## Capítulo 2

# Estado del arte de la realidad aumentada en dispositivos móviles

### 2.1 Realidad Aumentada

La Realidad Aumentada (RA) es una línea de investigación que trata de incluir información generada por computador sobre el mundo real. Esta definición difiere de la Realidad Virtual (RV), pues en la RV únicamente hay información virtual. Ambos campos se centran en proporcionar al usuario un entorno 3D inmersivo, aunque la RV se centra en proporcionar un entorno virtual para el usuario y la RA en alterar el mundo real con información virtual. El entorno que nos rodea es complejo y nos brinda información abundante que es difícil de interpretar y simular, es por ello que los ambientes creados con realidad virtual pueden llegar a ser simples y con falta de información del entorno que pretenden modelar. Una ventaja de la realidad aumentada es que ese entorno rico en información no se altera, y en lugar de ello se amplía con conocimientos que retroalimentan la escena que se pretende representar.

No existe una definición única de RA, aunque han aparecido algunas definiciones en diversas publicaciones. Milgram (MILGRAM94) define la RA sobre la base de un continuo llamado *Continuo de Milgram* (ver figura 2.1). Un entorno virtual se considera como algo totalmente sintético en el que los usuarios están completamente sumergidos; el entorno real se considera el lado opuesto, integrado sólo por los objetos reales limitado por las leyes de la física.



Figura 2.1: Continuo virtual de Milgram



La Realidad Mixta se encuentra a lo largo del continuo de Milgram y representa todos los sistemas que explotan los elementos tanto del entorno real como el entorno virtual al mismo tiempo. Dentro de esta realidad mixta, podemos distinguir entre Realidad Aumentada y Virtualidad Aumentada, dependiendo de cuál es el entorno principal (real o virtual) y cuál es el entorno secundario, que sirve de apoyo. Cuando un sistema está cerca de la parte central del continuo se vuelve más arbitrario ya que no queda claro cuál es el entorno preponderante sobre el otro. La RA se encuentra más cerca del entorno real que para el entorno virtual en el continuo de Milgram (MILGRAM94). Por tanto, la RA puede ser vista como una versión extendida del entorno real, complementada por los objetos virtuales.

Azuma (AZUMA97) identifica tres características fundamentales para la RA y da su propia definición de un sistema de RA, sin estar determinada a un hardware específico:

- un sistema de RA debe combinar realidad y virtualidad
- un sistema de RA debe ser interactivo en tiempo real
- la registración debe ser en 3D

En la última definición se presenta uno de los problemas básicos en RA: los objetos virtuales en 3D y objetos reales deben mostrarse siempre alineados entre sí. Se pueden utilizar varias técnicas para seguir la posición de los objetos virtuales y realizar una correcta colocación de ellos (seguimiento basado en reconocimiento de patrones a través de marcadores) para proporcionar diferentes niveles de precisión. El hardware que requiere esta técnica es por lo general más caro, pero la solución adoptada es más profesional. Una alta precisión en la colocación de los objetos puede ser por ejemplo, en el campo de la cirugía, además de que el sistema deberá responder también con una frecuencia de actualización mayor.

## 2.2 Realidad Aumentada móvil

Las primeras configuraciones de hardware en RA y la mayor parte de las actuales se basan en los equipos de sobremesa estática con cámaras fijas o en HMD (dispositivo de visualización similar a un casco) con ordenadores portátiles incorporados. Si bien las configuraciones de hardware tales proporcionan un alto rendimiento y por lo general dejan las manos libres, estas soluciones también presentan graves inconvenientes (altos costos, el atractivo social baja y las limitaciones a la destreza de los usuarios se convierte en un problema), que les impiden llegar a un público amplio de usuarios no técnicos. Un ejemplo del impacto de las diferentes soluciones de hardware en la RA se puede ver en la figura 2.2.



**Figura 2.2:** impacto en las diversas soluciones: mochila portátil con HDM, Tablet PC, PDA, Smartphone.

Las desventajas de los sistemas de sobremesa y los basados en mochila-HDM son especialmente evidentes cuando la RA se explota en computación móvil. Un sistema de RA móvil puede ayudar a los usuarios en cualquier lugar siempre que sea necesario, por lo tanto es un área de investigación

especialmente convincente. Cuando pensamos en movilidad, los dispositivos portátiles de mano (por ejemplo, Ultra Mobile PCs, PDAs, teléfonos inteligentes y consolas de juegos portátiles) aparecen como soluciones válidas para las aplicaciones de RA móvil. Dado que el presente proyecto ha sido diseñado y desarrollado para una plataforma portátil, este capítulo se centra principalmente sobre este tipo de dispositivos.

Hasta hace algunos años, los proyectos dirigidos de RA en dispositivos portátiles utilizaban éstos como front-end para la muestra de resultados, mientras que todos los cálculos se realizaban de forma remota y se retransmitían de nuevo a los dispositivos (aprovechando la red inalámbrica u otras conexiones). Los dispositivos móviles han crecido recientemente en el poder de computación y también en el procesamiento de gráficos 3D, sobre todo gracias a la introducción de procesadores de gráficos integrados (GPU), y además integrando las últimas capacidades inalámbricas y cámaras (debido a la presión comercial de las empresas de comunicación). El consumo de energía y el almacenamiento de ésta en las baterías no se espera que mejore mucho en los próximos años, luego la investigación actual se centra en incorporar procesadores especializados (como GPU codificadores de vídeo, decodificadores, procesadores vectoriales de punto flotante) y procesadores programables (FPGAs) para mejorar el rendimiento para ciertas clases de aplicaciones manteniendo al mismo tiempo unos límites aceptables de rendimiento (por ejemplo, mediante la desactivación de algunos procesadores)

Si bien en los enfoques iniciales era necesario el apoyo de hardware con más capacidad de cómputo debido precisamente a la baja capacidad de cómputo disponible en los viejos dispositivos, los avances de hardware han permitido un uso de dispositivos portátiles como plataformas de RA independientes. Tener sistemas autónomos es muy importante para la escalabilidad de las aplicaciones ya que cuando el cómputo se realiza localmente en cada dispositivo, la sobrecarga en el servidor se reduce considerablemente, o incluso se puede quitar porque los clientes no necesitan comunicarse entre sí o porque pueden comunicarse mediante otros

mecanismos. A pesar de las capacidades de movilidad y el nuevo potente hardware de estos dispositivos, estos tienen también algunas desventajas inherentes. Aunque los nuevos dispositivos vienen con hardware integrado para la aceleración 3D de gráficos, los altos consumos de energía hacen imposible que puedan competir con la calidad gráfica alcanzada por los sistemas de escritorio. Además los dispositivos portátiles no suelen estar equipados con una unidad de punto flotante de procesamiento (FPU) y por lo tanto sólo son capaces de realizar cálculos de punto fijo en el hardware, mientras que los cálculos de punto flotante se emulan en programas (lo que es hasta 50 veces más lento si se compara a la aplicación de hardware de punto flotante). Esto hace prácticamente imposible utilizar algoritmos fuertemente basados en cálculos de punto flotante. En los últimos años esto ha cambiado, ya que cada vez más estos dispositivos integran hardware con más capacidades. Otro factor a tener en cuenta es el tamaño de la pantalla y el estrecho campo de visión que limitan el tipo de interacción con el usuario. La técnica más utilizada por los usuarios es la denominada lente mágica. Al adoptar el enfoque de la lente mágica con el dispositivo este se está empleando como una lente para aumentar una parte del mundo real con información virtual. Una de las principales características de este enfoque es que son necesarios muchos movimientos con la cámara, si el usuario desea examinar un entorno de RA relativamente grande, porque sólo una pequeña parte de la escena se puede ver de una vez, lo que puede ser un inconveniente (por ejemplo, en una aplicación que requiere que el usuario tenga una visión global del entorno), o también una ventaja (por ejemplo, si la exploración del entorno es fija).

Otra desventaja en el desarrollo de sistemas de RA para dispositivos móviles es la falta de software, aunque algunos dispositivos recientes han abandonado las soluciones propietarias a favor de sistemas operativos estandarizados (por ejemplo, Microsoft Windows, Symbian, Linux, iPhone OS, Android) y bibliotecas de desarrollo (por ejemplo, Windows Mobile, OpenGL ES, OpenVG, ES OpenSL, OpenMAX6).

Puede parecer que el desarrollo de sistemas de RA para dispositivos móviles no pueda tener éxito debido a los múltiples factores que impiden que las aplicaciones funcionen tan bien como en equipos de sobremesa. En la tesis de Wagner (WAGNER07), se discuten tres hipótesis, que son demostradas a lo largo de ella, y que reafirmamos en este proyecto.

*La RA en dispositivos móviles puede funcionar tan bien como en los ordenadores de sobremesa, a pesar del hecho de que los teléfonos son menos potentes, tienen pantallas pequeñas y menos capacidades de entrada para el usuario.*

*La utilización de teléfonos móviles permite que se desarrollen más sistemas de RA debido al bajo costo de estos dispositivos.*

*El uso de los teléfonos móviles es ampliamente conocido por los usuarios, luego es más recomendable para usuarios comunes que el utilizar Tablet PC o HDM.*

## 2.3 Antecedentes

El presente apartado tiene como objetivo el realizar una breve retrospectiva sobre la historia de la RA, haciendo especial hincapié en los hitos en investigación y aplicaciones que han hecho que existan sistemas de RA para dispositivos móviles, aunque algunos puntos son compartidos por la RA para los equipos de sobremesa. Este breve resumen que se muestra a continuación ha sido extraído de (HMAR09).

**1968.** Sutherland (SUTHERLAND68) crea el primer sistema de RA, que es también el primer sistema de realidad virtual. Utiliza un HDM seguido por dos mecanismos diferentes para conseguir los seis grados de libertad: un tracker mecánico y un programa de seguimiento por ultrasonidos. Debido al limitado poder de procesamiento de los ordenadores en aquella época, sólo se podían crear dibujos sencillos alámbricos para mostrar en tiempo real.

**1992.** Caudell y Mizell (CAUDELL92) acuñan el término “realidad aumentada”, para referirse al hecho de añadir información generada por ordenador al mundo real. Caudell y Mizell discutieron acerca de las ventajas de la RA frente a la RV.

**1992.** IBM introduce el primer Smartphone, el IBM Simon Personal Communicator. El teléfono tenía 1 Megabyte de memoria y una pantalla en blanco y negro con 160x293 píxeles de resolución.

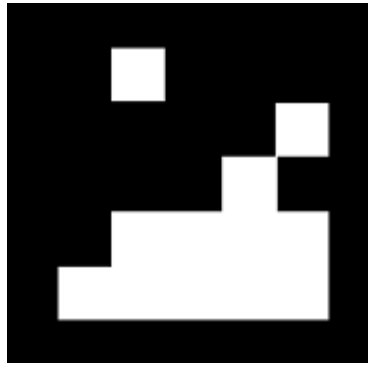
**1994.** Milgram y Kishino (MILGRAM94) describen en su “Taxonomía de la Realidad Mixta” el conocido término del continuo de Milgram (Reality-Virtuality Continuum). Engloba desde el mundo real hasta la realidad virtual, pasando por diferentes etapas, entre ellas la RA.

**1995.** Rekimoto y Katashi (REKIMOTO95) crean NaviCam. NaviCam utiliza una estación de trabajo y tiene una cámara montada que se utiliza para el seguimiento óptico. El equipo detecta los marcadores codificados en la imagen de la cámara en vivo y muestran información directamente sobre la secuencia de vídeo.

**1996.** Rekimoto (REKIMOTO95) presenta los marcadores de matriz 2D (cuadrados con forma de código de barras, ver figura 2.3), uno de los primeros sistemas de marcadores para permitir el seguimiento de la cámara con seis grados de libertad.

**1997.** Ronald Azuma (AZUMA97) presenta el primer estudio sobre RA. En su publicación, Azuma describe los aspectos más relevantes de la RA, identificada por tres características principales:

- Combina una escena real con objetos virtuales
- Interactiva en tiempo real
- Registración en 3D



**Figura 2.3:** Marcador de matriz 2D

**1999.** Kato y Billinghurst presentan ARToolKit (ART99), una librería de seguimiento con seis grados de libertad, utilizando fiduciales cuadrados (marcadores) y una plantilla para el reconocimiento de patrones. ARToolKit está disponible como código abierto bajo la licencia GPL y es todavía muy popular en la comunidad RA.

Hollerer et al. (HOLLERER99) presentan un sistema de RA móvil que incluye HDM y un tablet pc donde se captura la información del exterior (vídeo) y la envía a otro sistema, normalmente instalado en un puesto fijo.

**2000.** Julier et al. (JULIER00) presentan BARS (Battlefield Augmented Reality system). El sistema consiste en un HDM con backpack (mochila con ordenador y periféricos). Utiliza antena GPS y sensores de aceleración (actuales acelerómetros) para visualizar información en un hipotético campo de batalla, tal como localizaciones de estructuras e infraestructuras, así como posibles emboscadas de enemigos.

**2001.** Freund et al. (FRUEND01) presentan AR-PDA, un prototipo para construir sistemas de RA sobre PDA's. La idea básica es utilizar dichas PDA's junto con este sistema para interactuar con el hogar, por ejemplo, mostrar cómo funcionan algunos electrodomésticos del hogar.

Reitmayr y Schamassteig (REITMAYR01) presentan un sistema de RA móvil, multiusuario y colaborativo. La idea del sistema multiusuario es la de compartir

el mismo espacio real combinando los objetos virtuales e incrementando la interacción del usuario.

**2002.** Kalkusch et al. (KALKUSCKH02) presenta una aplicación para guiar al usuario a través del interior de un edificio hacia un destino particular dentro del mismo edificio. El sistema superpone un modelo en alámbrico de la estructura del edificio según se va avanzando por él, todo ello sobre un HDM. Utiliza marcadores de ARToolkit, captados y reconocidos a través de la cámara incluida en el conjunto de la aplicación, como se aprecia en la figura 2.4.



**Figura 2.4:** Aplicación ideada por Kalkusch et al.

**2003.** Wagner y Schmalsteig (SCHMALSTIEG03) crean un sistema de RA con el objetivo de servir como guía de interiores para un dispositivo tipo PDA. La aplicación provee al usuario un entorno aumentado con información del destino a llegar. Para este sistema no se necesitan dispositivos externos para su uso, luego es un sistema totalmente autónomo e independiente.

**2004.** Mohring et al. (MOHRING04) presentan un sistema para el posicionamiento con marcadores 3D en teléfonos móviles (figura 2.5), soportando la detección y la diferenciación de diferentes marcadores 3D, así como una correcta integración de gráficos 3D en la captura de vídeo.





**Figura 2.5:** Ejemplo de marcador 3D

Rohs y Gfeller (ROHS04) presentan Visual Codes, un sistema de marcadores 2D para teléfonos móviles. Estos marcadores pueden utilizarse sobre objetos físicos para superponer información virtual sobre dicho objeto.

**2005.** Henrysson (HENRYSSON05) porta ARToolkit para poder ejecutarlo en el sistema operativo Symbian. Basado en esta tecnología, presenta AR-Tennis, la primera aplicación de RA colaborativa para teléfonos móviles.

**2006.** Reitmayr et al. (REYTMAR06) presentan un modelo basado en un sistema de seguimiento híbrido de RA en entornos urbanos. Esto permite tomar, en tiempo real, captura de vídeo sobre en un dispositivo tipo PDA, para la visualización de la escena. El sistema combina diferentes dispositivos externos.

**2008.** Mobilizy lanza Wikitude (WIKITUDE09), una aplicación que combina el GPS y la brújula digital para mostrar datos de la wikipedia sobre lugares u objetos. Wikitude World Browser está desarrollado para el sistema operativo Android, implantado actualmente en muchos teléfonos móviles.

**2009.** Kimberly Spreen et al. (ARHH10) desarrollan ARhrrr! (figura 2.6), el primer videojuego de RA con una calidad gráfica parecida a los juegos comerciales. Esta aplicación utiliza el kit de desarrollo Tegra de Nvidia, optimizado para las GPU's actuales. Todo el procesamiento se realiza en la GPU, salvo en referido al posicionamiento, haciendo que la aplicación funcione con un alto ratio de frames por segundo.



**Figura 2.6:** ARhrr! Primer videojuego con gráficos comerciales

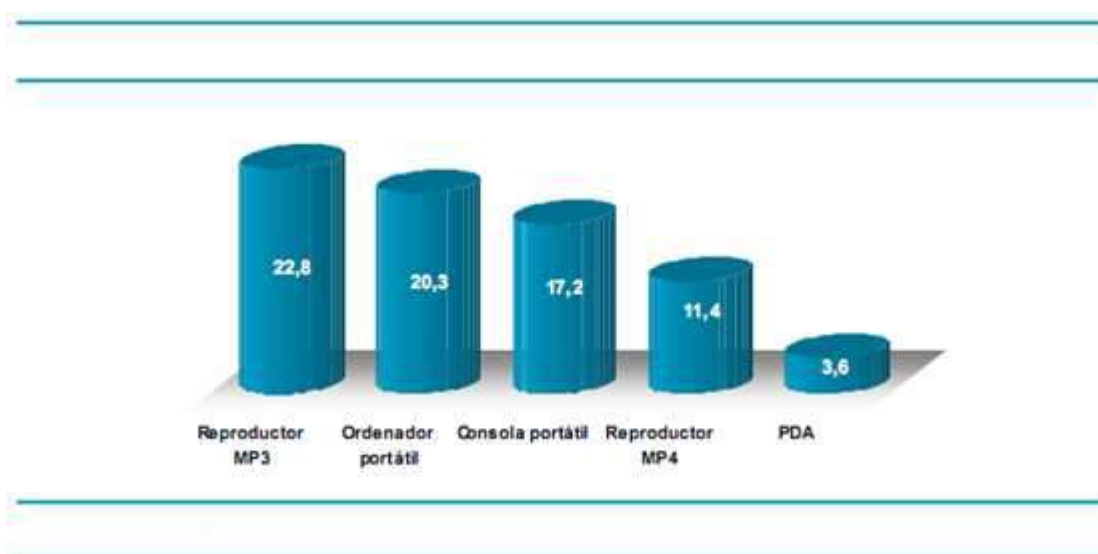
## 2.4 Dispositivos portátiles

En este apartado hablaremos acerca de la situación actual de los dispositivos portátiles, incluidos los móviles, en cuanto a integración en el mercado se refiere. En la segunda parte se expondrán cuáles son los dispositivos actuales que permiten adoptar sistema de RA, comentando algunas de sus características principales.

### 2.4.1 Situación del mercado

La constante evolución tecnológica nos permite disponer de dispositivos móviles con unas prestaciones cada vez mayores y a unos costes más competitivos, haciendo posible la integración de estos dispositivos al gran público. Según el estudio de la consultora *Multimedia Intelligence* (MULT08), en 2008 se vendieron en el mundo unos 300 millones de teléfonos móviles multimedia (se incluyen PDA, SmartPhone, etc.), entendiendo como multimedia aquellos dispositivos que disponen de cámara, tanto para realizar fotografías como para capturar vídeo. Por último destacar que en el estudio comentan que la alta competitividad entre operadoras hace que la evolución

tecnológica de los dispositivos se produzca más rápidamente, repercutiendo favorablemente en las prestaciones de las aplicaciones software (en nuestro caso para RA). Si bien dicho estudio se realizó a nivel mundial, dentro de nuestras fronteras los datos obtenidos auguran y demuestran que los dispositivos móviles son un mercado en constante expansión. En la siguiente gráfica (figura 2.7) extraída del informe anual del sector de las TIC en España (TIC08) observamos el porcentaje de población que posee los siguientes dispositivos: consolas portátiles y PDA.



**Figura 2.7:** Equipamiento individual. Tercer trimestre 2008

Por otro lado, la consultora de investigación de las tecnologías de la información Gartner, publicó en un estudio lanzado a mitad de 2009 que indicaba que las ventas de smartphones aumentaron un 29% respecto al mismo periodo del año anterior (Fuente obtenida de "El País" 12 de Noviembre de 2009). En otra de las fuentes consultadas, la medidora de mercados Canalys (CANALYS09), indica que se ha producido un incremento del 4% en la venta de teléfonos inteligentes en el tercer trimestre de 2009 con respecto al año anterior (a nivel mundial).

Estos datos apuntan a que la penetración de dichos dispositivos entre los usuarios permitirá disponer de dispositivos móviles que sean capaces de soportar aplicaciones de realidad aumentada, fomentándose así el desarrollo

de las mismas y permitiendo su expansión entre el público general y no sólo entre sectores profesionales determinados e investigadores.

## 2.4.2 Tipos de dispositivos

El desarrollo de RA para dispositivos portátiles se ha venido construyendo sobre los llamados HDM, que incorporando periféricos tan dispares como cámaras de vídeo, dispositivos de visualización (HDM) y un "backpack", que constituían los elementos necesarios para poner en funcionamiento un sistema de RA (ISWC97), pero estos sistemas tienen restricciones de peso, tamaño y coste. Con la aparición de dispositivos más reducidos y con capacidades de cálculo mayores la RA basada en dispositivos móviles ha logrado expandirse y acercarse a un público más general.

A continuación expondremos brevemente a modo de introducción alguna de las ventajas e inconvenientes de los diferentes dispositivos móviles que soportan aplicaciones de RA.

Dispositivo	Ventajas	Inconvenientes	
HDM (Head Mounted Display) + PC	<ul style="list-style-type: none"> <li>- Mayor inversión en la escena</li> <li>- Campo de visión grande</li> <li>- Potencia de cálculo</li> </ul>	<ul style="list-style-type: none"> <li>- Dispositivos caros</li> <li>- No están extendidos</li> </ul>	
Tablet PC	<ul style="list-style-type: none"> <li>- Capacidad de procesamiento</li> <li>- Resolución de pantalla</li> <li>- Buena interacción con el usuario</li> </ul>	<ul style="list-style-type: none"> <li>- Pueden requerir de una cámara externa</li> <li>- Poco manejables</li> <li>- Consumo de batería</li> </ul>	
Ultra Mobile PC	<ul style="list-style-type: none"> <li>- Capacidad de procesamiento</li> <li>- Fácil transporte</li> <li>- Resolución de pantalla</li> </ul>	<ul style="list-style-type: none"> <li>- Pueden requerir de una cámara externa</li> <li>- Caros con respecto a PDA/Smartphone</li> <li>- Consumo de batería</li> </ul>	
PDA	<ul style="list-style-type: none"> <li>- Reducido tamaño</li> <li>- Prestaciones suficientes para RA</li> <li>- Conectividad</li> </ul>	<ul style="list-style-type: none"> <li>- Pueden requerir periféricos externos</li> <li>- Tienen a converger hacia los Smartphone</li> </ul>	
Smartphone	<ul style="list-style-type: none"> <li>- Disponen de sensores y dispositivos internos como: cámara, gps, etc.</li> <li>- Popularización en auge</li> <li>- Conectividad</li> </ul>	<ul style="list-style-type: none"> <li>- Dispositivos con menos potencia de cálculo hoy en día (en aumento)</li> </ul>	
Videoconsola Portátil	<ul style="list-style-type: none"> <li>- Prestaciones similares o superiores que las PDA/Smartphone (llevan FPU/GPU)</li> <li>- Buen rendimiento en gráficos</li> <li>- Muy extendidas</li> </ul>	<ul style="list-style-type: none"> <li>- Su funcionalidad está limitada al entretenimiento</li> <li>- Requieren cámara externa</li> <li>- Plataformas propietarias</li> </ul>	

### **Tablet-PC**

Con el fin de utilizar dispositivos móviles más potentes para utilizar en RA, se han introducido nuevos dispositivos tales como las Tablet-PC. Un Tablet-PC es un notebook con pantalla táctil que permite una alta interacción entre el usuario y el sistema. El uso de estos dispositivos elimina los inconvenientes de hardware y de software de dispositivos más pequeños y los problemas ergonómicos de los HDM con backpack.

### **UMPC**

Otro de los dispositivos que se han venido utilizando en RA móvil son los UMPC. Desarrollados por Microsoft™, Intel™, and Samsung™, entre otros, los UMPC son básicamente pequeños PC con el sistema operativo Microsoft Windows XP y disponen de pantalla táctil. Un gran número de aplicaciones han utilizado estos dispositivos para probar su funcionamiento, pero todo apunta a que los potenciales usuarios de RA móvil no utilizarán los UMPC, sino los Smartphone.

### **PDA**

Antes de la introducción de los UMPC y los SmartPhones con CPUs computacionalmente más rápidas, las PDA han sido la única alternativa para los investigadores de RA. Las PDA ofrecen pantallas relativamente grandes, conectividad sin cables, sistema de GPS, etc. Sin embargo, no tienen capacidad para procesar gráficos 3D y no disponen de unidad de coma flotante. Ejemplos de aplicaciones pueden ser (SCHMALSTEIG07) (TIMEWARP07). La mayoría de estas aplicaciones han sido desarrolladas en el sistema operativo Microsoft Windows Mobile™.

## **Smartphone**

Actualmente los teléfonos móviles son los dispositivos más usados. Teniendo en cuenta las limitaciones que presentan, estos dispositivos cuentan cada vez con mayores prestaciones. Pueden servirse de un servidor donde realizar los cálculos con más carga computacional y luego enviar los resultados al Smartphone para la visualización final.

Los sistemas operativos más comunes para los Smartphone son:

- Windows Mobile (Windows Phone)
- Symbian OS
- Iphone OS
- Android
- BlackBerry OS

## **Videoconsola portátil**

En la tabla anterior se han expuestos su características principales. Estos dispositivos no han sido empleados en el desarrollo e investigación de aplicaciones para RA, pero sí que existe software comercial que hace uso de esta tecnología. Dos ejemplos de ello son los videojuegos Invizimals y GhostWire.

Todos estos dispositivos presentan diferentes rendimientos debido a que su hardware es notablemente diferente en cada caso. También es posible que un mismo sistema de RA móvil utilice dos o más dispositivos como los anteriores, como es el caso de TimeWarp (TIMEWARP07), un juego desarrollado en el instituto tecnológico Fraunhofer FIT, que utiliza un HDM para la visualización de la información y un UMPC para realizar el rendering y los cálculos, además de mostrar otro tipo de información al usuario. El incorporar sensores externos como cámaras no siempre es posible, bien sea porque el dispositivo es demasiado pequeño o bien porque añaden peso y aparatosidad al dispositivo y no lo convierten en una buena solución móvil.

Otro aspecto a tener en cuenta es la velocidad del procesador. Esto no es un problema para los Tablet PC y los UMPC, donde nos encontramos CPU's

desde los 800 Mhz hasta los 2,2 Ghz que pueden llegar a tener los modelos más actuales. A partir de este punto y hasta finalizar este capítulo, nos centraremos únicamente en el hardware de los Smartphone, PDA y videoconsolas portátiles, porque en ellos el hardware juega un papel crucial. Estos tres dispositivos incorporan procesadores ARM o de similares características, y la velocidad del procesador suele situarse entre los 100 y los 600 Mhz y no siempre tiene unidad de coma flotante (FPU), lo que hace que las posibilidades en aplicaciones multimedia sean más reducidas y se tenga que emular por software estas operaciones, ralentizando todo el sistema. Aunque lo más frecuente es que estos dispositivos incluyan una GPU (Unidad de procesamiento gráfico) donde se realizan los cálculos para el rendering 3D, las operaciones en coma flotante, instrucciones matemáticas dedicadas para el renderizado de gráficos, operaciones gráficas (primitivas), etc.

La cantidad de memoria RAM también es un punto a tener en cuenta en los dispositivos tipo PDA/Smartphone/videoconsola portátil, ya que incrementa el coste de los dispositivos y consume batería, al tener que refrescarse constantemente. Suelen tener desde 32 Mbytes hasta los 300 MBytes.

Estos dispositivos requieren de cámaras externas o bien la pueden llevar integrada. Es esencial que se trabaje y de soporte a los formatos nativos de píxel para tener un alto rendimiento en la aplicación. No es recomendable la conversión entre distintos formatos porque en estos dispositivos hay grandes limitaciones en el ancho de banda de la memoria (WAGNER07).

Por últimos, haremos referencia a los tres aspectos más importantes a tener en cuenta en el hardware y que son de vital importancia para obtener buenos resultados en aplicaciones de RA móvil. Las conclusiones han sido extraídas de la tesis de D.Wagner (WAGNER07):

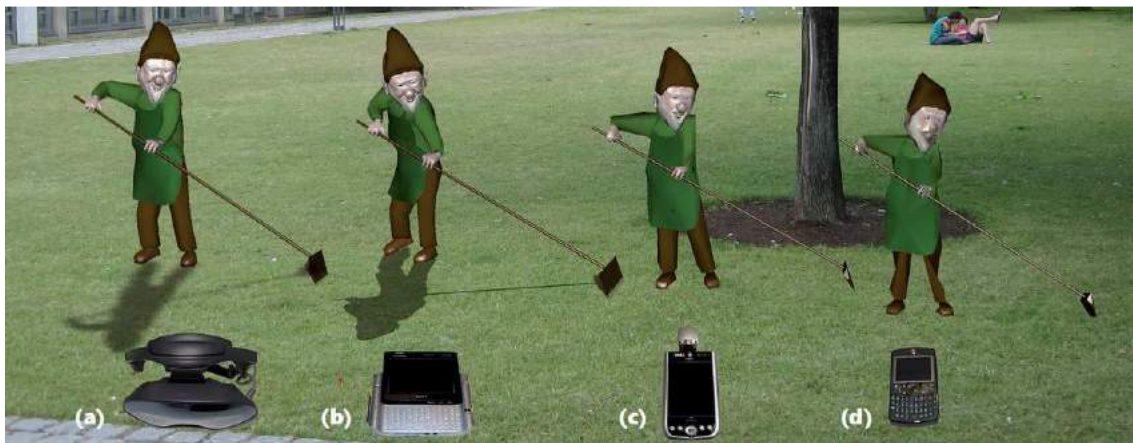
- A pesar de que la CPU ARM generalmente no tiene unidades de ejecución en paralelo, muchas operaciones como la lectura de la cámara o de la red de comunicación pueden ser aceleradas con



éxito usando multi-threading, porque son operaciones de E / S en vez de CPU.

- Algunos teléfonos de gama alta pueden ofrecer secuencias de vídeo en resolución de hasta 640x480 píxeles. En la práctica, estos vídeos están limitados a la calidad de la cámara. A diferencia de las cámaras de PC de alta calidad, sólo hay una mejora mínima en la calidad del seguimiento con respecto a cámaras con resoluciones menores. Las razones de esto son las lentes de baja calidad y unos sensores con altos niveles de ruido.

Podemos ver en la figura 2.8 una muestra de las capacidades actuales que nos ofrecen estos dispositivos:



**Figura 2.8:** Resultados según tipo de dispositivo

Los resultados visuales son mejores según el dispositivo empleado, siendo el que mayor calidad ofrece el HDM con un PC. Actualmente, las PDA han pasad a segundo plano en cuenta a hardware se refiere y son los Smartphone los que cada vez están contando con un hardware más potente, llegando a poseer algunos modelos hasta 1 GHz de frecuencia del procesador y avanzadas GPU.

## 2.5 Campos de aplicación

Este apartado tratará sobre los diferentes campos de aplicación existentes en el marco de la realidad aumentada móvil. Los campos de aplicación son diversos y sirven de apoyo a varias disciplinas y servicios: mantenimiento y reparación industrial, aprendizaje, entretenimiento, guía turística, localización GPS, publicidad, negocios, etc. Existen diversos campos que pese a ser empleados por sistemas de RA clásicos, no los hemos incluido en el proyecto por que los actuales dispositivos no pueden ofrecer toda la funcionalidad necesaria para el correcto funcionamiento, debido entre otros a la limitación de las pantallas, la capacidad de procesamiento y la incomodidad de manejar los dispositivos en determinadas situaciones. Es por ello que la RA para dispositivos móviles presenta menos campos de actuación, debido a las restricciones de los dispositivos y a las condiciones de los campos de actuación.

***Mantenimiento y reparación industrial:*** ante la complejidad de un montaje de un sistema industrial cualquiera, es posible que los técnicos encargados de ello requieran el uso de manuales o un estudio previo del montaje o de la reparación de dicho sistema. Una forma tecnológicamente más avanzada de afrontarlo es mediante la utilización de un sistema de RA móvil, donde el ingeniero o técnico pueda visualizar en pantalla la disposición de los objetos tratados y añadir dicha información sobre ellos, básicamente la esencia de la RA. Esto conseguiría aumentar la productividad, la precisión y la seguridad del personal encargado. Un ejemplo de ello podría ser el montaje de un motor de combustión, como muestra la figura 2.9, donde el mecánico visualiza en su dispositivo HDM cómo se ha de desmontar una determinada pieza. También podemos encontrar, dentro de este ámbito, aplicaciones destinadas a la reparación de aeronaves y al sector de la automoción. Importantes empresas del sector automovilístico han apostado por sistemas de RA sobre HDM, con el fin de dar soporte al montaje y reparación de vehículos.



**Figura 2.9:** Aplicación de RA en reparación industrial

**Entretenimiento:** este es un campo donde la RA presenta una mayor interacción con los usuarios. Con el avance del hardware y ante la búsqueda de nuevas formas de entretenimiento, la RA se ha hecho un hueco en este campo, ya que presenta interesantes novedades y nuevas formas de entretenimiento, desvinculándose de los campos donde se ha venido aplicando tradicionalmente. El campo del entretenimiento es bastante amplio, pero distinguimos de entre todos los ejemplos de entretenimiento al del sector de los juegos (ver figura 2.10), por ser uno de los ejemplos más claros donde la RA móvil puede utilizarse.



**Figura 2.10:** Aplicación AR Tennis

**Aprendizaje:** las nuevas herramientas digitales hacen posible una mayor integración de los medios digitales en los métodos de enseñanza y aprendizaje. Muchos investigadores y educadores coinciden en que el uso de nuevas tecnologías hace que el interés de los alumnos y la participación activa de éstos aumenten. La RA usada en el aprendizaje persigue varios

objetivos claros: desarrollar sistemas para aprender de manera más rápida conceptos a partir de interacciones que puedan realizar los propios alumnos, conseguir un entendimiento más claro y profundo del proceso de aprendizaje humano, crear aplicaciones que permitan acelerar el proceso de aprendizaje, etc.

**Turismo:** el turismo es una de las fuentes de ingresos principales para muchas ciudades y países, sumado a la integración de nuevas tecnologías, hacen posible que la RA móvil esté presente en este campo. Los sistemas desarrollados varían en función de la finalidad de su uso pero todos tienen en común la propiedad de estar diseñados para ser utilizados como guía turística, añadiendo información en tiempo real del entorno (ver figura 2.11) y objetos que el usuario esté viendo en esos momentos. Es posible, que dado el elevado número de funcionalidades de las aplicaciones estudiadas más adelante, éste campo presente características comunes a otros, como pueden ser el entretenimiento, la publicidad, la neogeografía, etc.



**Figura 2.11:** Aplicación Archeoguide

**Neogeografía:** pese a ser un concepto relativamente nuevo, la neogeografía ha supuesto nuevos modelos de entendimiento y desarrollo cultural, creando un mundo más global. Podemos entender la neogeografía como un fenómeno social y de una nueva relación de las personas con el espacio físico, debido a la masificación de los mapas virtuales, el acceso a la anotación de éstos y el abaratamiento de dispositivos de posicionamiento tales como

el GPS. Estamos pues, ante un campo con unas posibilidades muy grandes, y con un rápido crecimiento de aplicaciones. Los elementos más comunes de la neogeografía son los mapas virtuales, la georeferenciación y la información añadida a los mapas digitales. La integración de todos estos elementos junto con el uso de dispositivos móviles que incorporen brújulas digitales y GPS, son los principales actores de la neogeografía en RA móvil. Para ilustrar y aclarar mejor este concepto, imaginemos que podemos visualizar información sobre un determinado sitio geográfico (monumento, calle, reserva natural, etc.) según vamos caminando por el entorno, con tan sólo enfocar la pantalla de nuestro dispositivo sobre el objeto o sitio donde queremos obtener información, como se muestra en la figura 2.12.

Principalmente estos sistemas se encuentran en el ámbito de la web y los ordenadores personales, pero como se comentó anteriormente, los nuevos dispositivos móviles brindan nuevas posibilidades de desarrollo, potenciando la RA móvil, y atreviéndose el autor de este proyecto a afirmar, que éste es actualmente uno de los campos más importantes y de más proyección de la RA móvil.

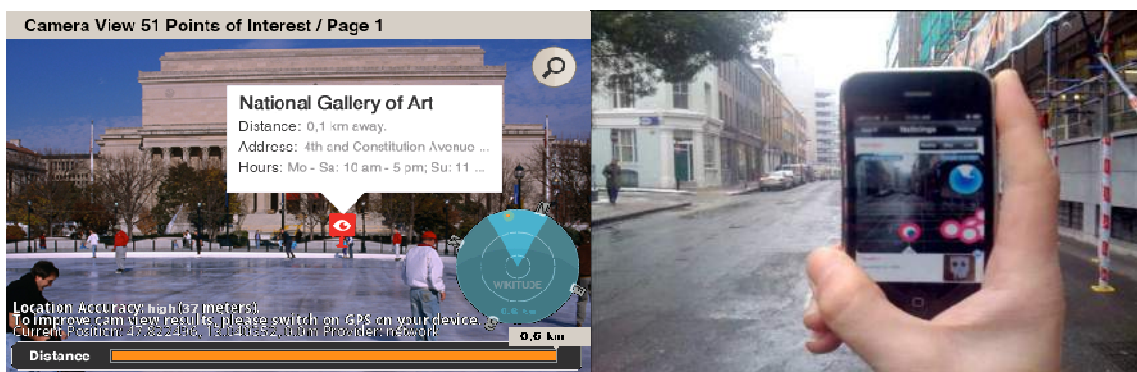


Figura 2.12: Aplicación Wikitude y Layar

**Publicidad:** la búsqueda incesante de nuevas fórmulas y soportes publicitarios sigue generando nuevas propuestas e innovadoras ideas que gracias al uso de las nuevas tecnologías suponen un avance en cuanto a conceptos, usabilidad y sus diferentes aplicaciones publicitarias. Aplicado al mundo de la publicidad, la RA puede aportar al consumidor una experiencia nueva y distinta al

combinar la realidad con imágenes y objetos virtuales que interactúan con el consumidor en tiempo real y en tres dimensiones. La verdadera proliferación está sujeta a las diferentes mejoras y avances de aquellos dispositivos que pueden convertirse en herramienta operativa capaz de procesar y ofrecernos información a través de la RA. Actualmente, la publicidad a través de la RA se puede encontrar en varias aplicaciones comerciales, aunque su uso es más extendido fuera de nuestro país.

**Negocios:** si bien es posible que este dominio no esté muy bien diferenciado para el contexto en el que está descrito, podemos ver mediante este supuesto que es un campo diferenciado y propio: imaginemos que somos un empresario que desea hacer su empresa más competitiva. Con un sistema de RA, podemos integrar información en la tarjeta de la empresa (a modo de marcador), permitiendo ver en la pantalla de nuestro dispositivo, información como datos de la empresa, algunos de sus productos estrella, anuncios, etc. Pensemos ahora, que en nuestra empresa queremos que los clientes puedan ver esta información de manera digital, pues de manera análoga podemos integrar información en los catálogos de la empresa para que capten la atención de los clientes de manera más activa. En esta misma línea, existe la denominada por algunos como "identidad aumentada", un tipo de RA donde no se utilizan los marcadores habituales, si no que se utiliza la cara de la persona con tal de mostrar información personal de ella, como datos personales, perfil de Facebook, Twitter, curriculum, etc.



## 2.6 Ejemplos de aplicaciones

### ***Mars. Touring Machine***

Nuestra primera aplicación a análisis es MARS (Mobile Augmented Reality Systems). Fue creada en 1997 por el laboratorio de Gráficos por Computador e Interfaces de Usuario de la Universidad de Columbia, y es el primer sistema de realidad aumentada que se utilizó en dispositivos móviles (MARS97). Se trata de un sistema formado por un HDM y un dispositivo tipo PDA (figura 2.13), que como ya comentamos, se trata del primer soporte hardware capaz de dar soporte a una aplicación de RA móvil. El objetivo principal de este prototipo era el de servir como guía de información en el campus de la universidad, mostrándonos información acerca de los edificios más importantes así como hechos relevantes que tuvieron lugar en él. Seleccionando mediante una interfaz gráfica en la PDA el edificio al que se quiere acceder a su información, el sistema se encarga de proporcionar los datos necesarios y visualizarlos en la PDA, mientras que en el visor del HDM se visualiza la localización de dicho edificio, además de su lejanía o proximidad, algún dato relevante sobre él, etc.



**Figura 2.13:** Aplicación Mars Touring Machine

Este proyecto fue el antecesor de los actuales sistemas de RA móvil en los campos de la neogeografía y el turismo. El laboratorio de investigación donde fue diseñado también creó otros prototipos parecidos pero siguiendo la misma filosofía: integrar la RA en unos dispositivos que eran cada vez más económicos y con un rendimiento cada vez mayor. Los sistemas posteriores también utilizaron los dispositivos de posicionamiento GPS y se prestó especial interés en el renderizado de gráficos, aunque debido a la aparatosidad de aquellos dispositivos los proyectos sólo quedaron en prototipos, aunque tienen el mérito de ser los pioneros de la RA para dispositivos móviles.

### **ARMAR**

El proyecto ARMAR (Realidad Aumentada para el mantenimiento y reparación) explora el uso de RA para ayudar en la ejecución de las tareas de procedimiento en el mantenimiento y la reparación mecánicas, industriales, etc. El objetivo principal de investigación de este proyecto es determinar en qué medida los gráficos por ordenador generados en tiempo real, superpuestos en el equipo físico como información adicional (ver figura 2.14), pueden ayudar a mejorar la productividad, la precisión y seguridad del personal de mantenimiento. La virtualización del usuario y el entorno de mantenimiento permiten a los colaboradores situados fuera del entorno supervisar y ayudar con las reparaciones. Además, la integración de las bases de conocimiento real del mundo con detallados modelos en 3D ofrece la posibilidad de utilizar el sistema como un simulador de mantenimiento y herramienta de formación. Este proyecto cuenta con el diseño y ejecución de prototipos de la integración de las últimas novedades en el seguimiento de movimiento, de dispositivos móviles, redes inalámbricas, modelado en 3D, y de las tecnologías de interfaz hombre-máquina (HENDERSON07).





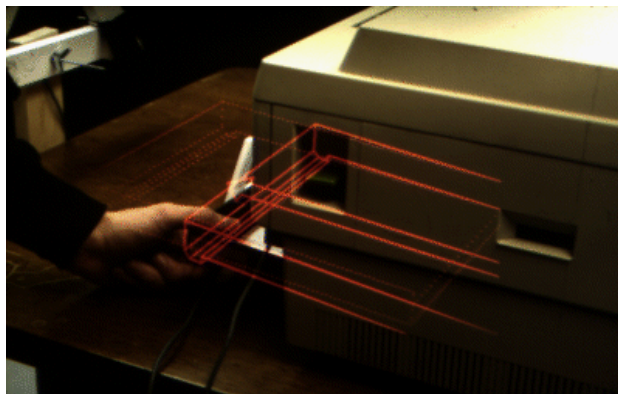
**Figura 2.14:** Aplicación ARMAR

Los beneficios de este tipo de sistemas han sido comprobados por el laboratorio de investigación donde fue creado, el laboratorio de Gráficos por Computador e Interfaces de Usuario de la Universidad de Columbia. El prototipo fue instalado en la cabina de un vehículo blindado estadounidense para facilitar las tareas de reparación mecánicas más rutinarias, tales como la instalación y la eliminación de elementos de sujeción y de las luces indicadoras, cables de conexión, etc. Los mecánicos encontraron este sistema cómodo e intuitivo para realizar sus tareas, al no tener que consultar los manuales en papel y de tener información del proceso de montaje en todo momento.

## **KARMA**

Otra de las primeras aplicaciones que se diseñaron fue KARMA (Realidad Aumentada basada en el conocimiento para la asistencia técnica), un prototipo (FEINER93) que utiliza un HDM para explicar el mantenimiento que un usuario debe realizar a una impresora láser (figura 2.15). Este prototipo no utilizaba marcadores matriz 2D, si no unos pequeños triángulos de plástico para hacer de marcadores, permitiendo así al sistema saber la posición y orientación de la impresora y del usuario. El objetivo del sistema es mostrar al usuario las distintas partes de la impresora, siendo el sistema capaz de saber si una parte está ocultada por otro objeto, con respecto a lo que visualiza el

usuario. Si esto se produce, se mostrará su modelo virtual en la posición correspondiente.



**Figura 2.15:** Aplicación KARMA

### **MR.VIRTUOSO**

Mr. Virtuoso (WAGNER07) es un juego educativo de colaboración, diseñado para mostrar las posibilidades de la RA en dispositivos móviles. Esta aplicación fue diseñada por la Universidad Tecnológica de Graz.

El objetivo del juego es ordenar una colección de obras de arte de acuerdo a su fecha de creación a lo largo de una línea temporal establecida en una pared (izquierda = anterior, derecha = posterior). Cada marca en la línea temporal representa una de las obras de arte, que sólo son visibles a través de la aplicación y la RA en la PDA del jugador. Inicialmente, las obras de arte están ordenadas aleatoriamente. El jugador puede recoger cualquier obra de arte con su PDA, haciendo clic sobre la ilustración en la pantalla y colocarla en una posición libre haciendo clic en un de las marcas libres en la pantalla. Inicialmente todas las marcas están ocupadas por una obra de arte, lo que requiere de dos o más jugadores para cooperar y reorganizar la secuencia de obras de arte. Es por ello que es un juego de colaboración. Mientras que una obra de arte seleccionada se encuentra en la PDA, el jugador puede acceder a su información, mostrando los datos más importantes al respecto. Además el jugador también puede consultar al experto virtual en historia del arte, Mr. Virtuoso. Para conseguirlo, hay que

colocar la obra de arte en su escritorio, que contiene otra de las marcas. Cuando una obra de arte se coloca en una posición correcta, ésta ya no se puede mover de nuevo, lo que facilita el juego a aquellos jugadores que quieran completar la secuencia correcta de obras.



**Figura 2.16:** Interacción de usuarios con Mr. Virtuoso

Este juego incluye una cantidad comparativamente grande de contenidos multimedia, pese a ser creado para dispositivos móviles. Tiene una selección total de 20 obras de arte, pudiéndose seleccionar un subconjunto para iniciar una nueva partida. El juego cuenta con texturas, modelos animados 3D, multimedia, material de referencia y una narración de audio en tres idiomas (Inglés, alemán, español).

### El tren invisible

El tren invisible (WAGNER07) es la primera aplicación multi-usuario de RA para dispositivos móviles. Este sistema únicamente utiliza como hardware un dispositivo móvil, a diferencia de otros proyectos que se lanzaron al mismo tiempo (2005) que utilizaban tanto dispositivos móviles como servidores para realizar la mayoría de los cálculos (por ejemplo, de procesamiento de gráficos). Es por tanto un adelanto considerable en el campo de la RA móvil, que habitualmente venía utilizando otro tipo de equipos no móviles para realizar las tareas que requerían más potencia de cálculo. El tren invisible es un juego en el que los jugadores controlan los trenes virtuales en un raíl de madera en miniatura. Estos trenes virtuales sólo son visibles para los jugadores a través de su PDA (ver figura 2.17), ya que no existen en el mundo físico.



**Figura 2.17:** El tren invisible

Los jugadores pueden interactuar con el ambiente de juego mediante el ajuste de las vías y de la velocidad de los trenes virtuales (ver figura 2.18). El estado actual del juego se sincroniza entre todos los participantes a través de redes inalámbricas. El objetivo común del juego es impedir que los trenes choquen, virtualmente hablando.

Al igual que Mr. Virtuoso, esta aplicación fue desarrollada con el software Studierstube de la Universidad Tecnológica de Graz. Utiliza una arquitectura basada en componentes y ha sido diseñado para acelerar la tarea de desarrollar y desplegar aplicaciones de RA de colaboración en dispositivos móviles.

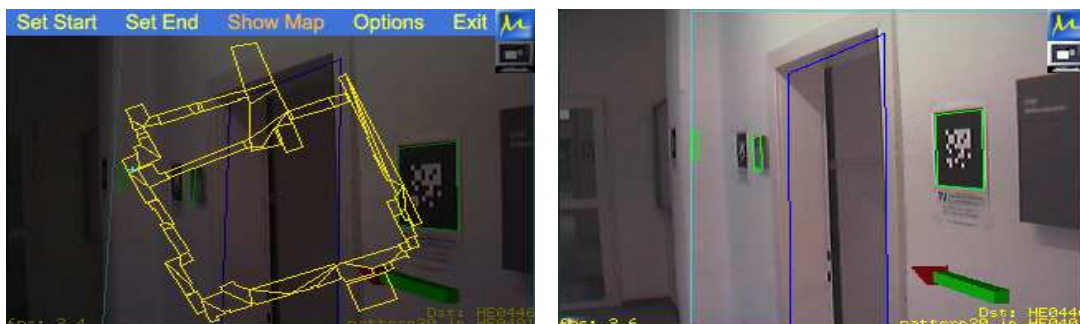


**Figura 2.18:** Aplicación multi-usuario El tren invisible

### **SIGNPOST 2003 & 2007**

La siguiente aplicación a tratar, SignPost, es un navegador de RA para facilitar al usuario a moverse dentro de un edificio desconocido, ya que se le muestra por medio de flechas el destino hacia donde quiere dirigirse. Incluye otras características tales como la visualización en alámbrico de ciertas partes del edificio y un mapa en 2D de la planta del edificio donde se encuentre, como se aprecia en la figura 2.19. Para dicha tarea, SignPost utiliza marcadores instalados en las paredes del edificio, utilizando el software Studierstube para el reconocimiento y obtención de la matriz de transformación de los objetos a visualizar.





**Figura 2.19:** SignPost 2003

Debido a que la potencia de cálculo de los dispositivos móviles de hace unos años eran inferiores a los actuales, el sistema podía funcionar en dos modos: realizar el procesamiento en el cliente local (PDA) o en el servidor, si éste se encontraba disponible, ofreciendo más rápidamente los resultados así como una mejor calidad en los resultados.

La siguiente versión de este proyecto fue Signpost2007, una guía de localización diseñada para ayudar a los asistentes de una conferencia de Microsoft para localizar lugares. Permite al usuario obtener información sobre una conferencia o programa de un evento, mostrando al usuario en el mapa cada sesión que se lleva a cabo y le dice al usuario cómo llegar desde su ubicación actual. Esta aplicación también utiliza el software Studierstube Tracker, y al igual que el anterior proyecto, fue desarrollado en la Universidad Tecnológica de Graz.

### **The Louvre - DNP Museum Lab (LDML)**

El proyecto LDML (MIYASHITA08) está desarrollado para ser usado en el museo del Louvre, y con la finalidad de mostrar información de las obras de arte expuestas (o incluso una representación gráfica en 3D de la obra, ver figura 2.20) así como también pretendía ser una guía de visita para algunas secciones del museo, concretamente el área de arte islámico. El sistema utiliza un backpack y una pantalla LCD para la visualización de resultados, y el software utilizado es Unifeye SDK, de la compañía Metaio.

Esta aplicación representa uno de los primeros pasos para extender y dar a conocer la RA entre el gran público. De las entrevistas de los usuarios que utilizaron el sistema, se desprende que la utilización de la RA fue muy satisfactoria para ellos, debido entre otras razones, a que el sistema les ayudó a percibir mejor los detalles de las obras de arte, tales como el reverso de las piezas cerámicas expuestas. También incluye un sistema de navegación para orientarse dentro del museo, siendo aún más satisfactoria la visita al museo.



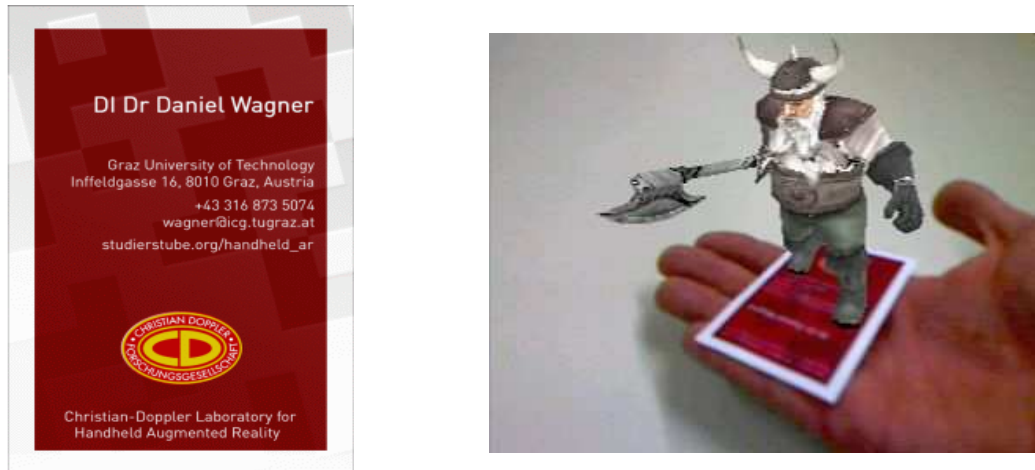
**Figura 2.20:** The Louvre DNP Museum LAB

Con este proyecto se puede observar cómo la RA puede contribuir no sólo al entretenimiento de las personas, si no a su formación cultural. Más adelante veremos más ejemplos donde la barrera que separa el entretenimiento del aprendizaje se hace más difusa, con todos los beneficios que ello puede aportar a las personas.

### **AUGMENTING BUSINESS CARDS – TU GRAZ**

La tarjeta de negocio aumentada es un prototipo de marcador desarrollado en la Universidad Tecnológica de Graz (figura 2.21). Habitualmente los marcadores vienen utilizando unas marcas en blanco y negro así como una frontera que rodea al marcador (ID, DataMatrix, plantilla, etc.). Este nuevo marcador, se ha adaptado para que pueda ser utilizado como tarjeta de negocios, donde mediante el software apropiado, mostrar información asociada a la persona que lo entrega. La imagen capturada es

posteriormente pasada al blanco y negro para poder procesarla. Con este ejemplo se pretende demostrar que es posible utilizar otro tipo de marcadores con un aspecto visual diferente de los habituales marcadores.



**Figura 2.21:** Tarjetas aumentadas

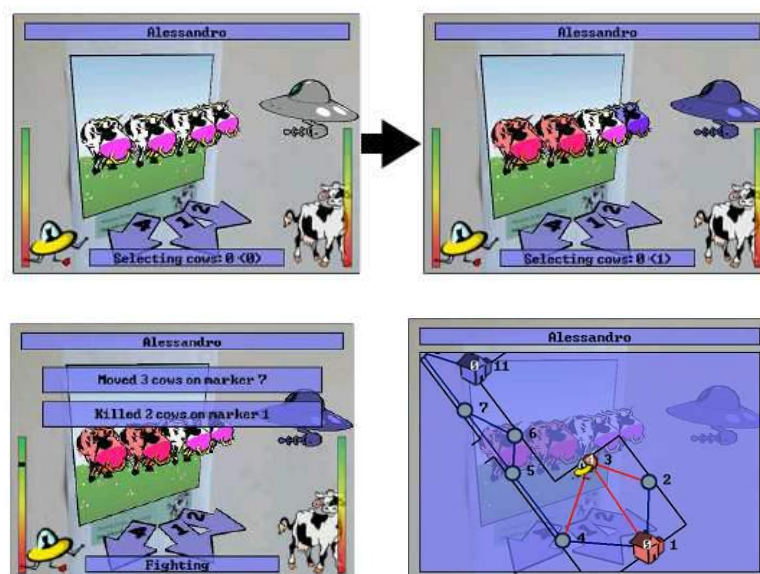
### **COWS VS. ALIENS 2007**

El sistema de RA Cow Vs. Aliens es el resultado de una tesis leída en la Universidad de Udine, posteriormente el desarrollo fue mejorado en la Universidad Tecnológica de Graz (MULLONI07). Es un juego multi-jugador donde los jugadores se dividen en dos equipos que compiten. La configuración de juego y el entorno de juego se divide en un grupo de lugares: los pastizales, donde las vacas están pastando tranquilamente, y dos lugares especiales donde se encuentra un establo para cada equipo. Estos lugares adyacentes están conectados entre sí y forman un grafo. Las vacas no son conscientes del hecho de que una invasión alienígena está ocurriendo y los ovnis están presentes en algunos de los lugares, dispuestos a disparar a cualquier vaca que trate de escapar de uno de los sitios mencionados antes (ver figura 2.22).

Para utilizar el sistema es necesario establecer una serie de marcadores colocados en las paredes. Estos marcadores forman un grafo cuyos nodos son lugares reales que están interconectados entre sí por una línea de visión una



relación de adyacencia física. El ambiente del juego virtual está mapeado en el mundo real, cada ubicación real (un marcador en la pared) representa una ubicación virtual del entorno del juego, mientras las líneas entre nodos se asignan en las interconexiones virtuales durante el juego. La interacción entre los jugadores y el ambiente del juego se basa en la interacción de los jugadores con el entorno: los jugadores se ven obligados a moverse físicamente en el entorno real para alcanzar ciertos lugares antes de que se pueda emitir cualquier acción en el juego virtual. Utilizando el software Studierstube ES, las entidades 3D del juego se superponen a los marcadores a través de la pantalla de los dispositivos móviles. Los elementos virtuales ayudan a dar a todos los jugadores la impresión de que están físicamente visitando el entorno de juego y los lugares virtuales.



**Figura 2.22:** Imágenes de Cows vs Aliens

El objetivo final del juego es salvar a las vacas, y estas sólo pueden ser consideradas seguras cuando se encuentran dentro de un establo: el objetivo de los jugadores es recoger las vacas de todo el espacio de juego y llevarlas sanas y salvas a la cuadra que pertenece a su equipo. Los jugadores pueden moverlas entre los lugares conectados (los pastos pueden contener un máximo de cuatro vacas). Las vacas deben tratar de evitar hacer frente a los ovnis, de otro modo no se salvarían. Cualquier ovni puede ser mandado por otro

jugador a disparar a todas las vacas que están presentes en un mismo lugar. Un objetivo secundario de los jugadores es, por tanto, evitar que las vacas sean atacadas por los ovnis. El ganador del juego es el equipo que guarda la mayoría de las vacas en el establo.

### **Situated Simulations**

Este proyecto está desarrollado en el departamento de Comunicaciones y Multimedia de la Universidad de Oslo (LIESTOL09), y pretende ser un proyecto englobado dentro del nuevo concepto de *meaningware* (interrelación entre hardware, software y usuarios). En esta aplicación, la percepción visual real del usuario se ve mezclada con los objetos 3D generados por el dispositivo móvil, como se observa en la figura 2.23. Para conseguir esto, la relación entre lo real y lo virtual se obtiene haciendo que el entorno 3D se adapte a la posición de la cámara y al movimiento del usuario, con tal de determinar la posición del entorno 3D. A medida que el usuario se mueve en el espacio real, la perspectiva del espacio virtual varía en consecuencia. Es por ello que se requieren dispositivos móviles con mucha capacidad de procesamiento, además de periféricos tales como el GPS, acelerómetro y brújula electrónica.



**Figura 2.23:** Imágenes de Situated Simulations

## Second Sight

Dentro del ámbito del aprendizaje nos podemos encontrar con Second Sight (CONNECTED10), una aplicación de RA para la videoconsola portátil Sony PSP. Este proyecto (desarrollado por la compañía ConnectED) provee a profesor y alumno contenido educacional interactivo. En su conjunto, Second Sight, a través de la RA, enriquece la calidad de la enseñanza ayudando al profesor a impartir las lecciones junto con material audiovisual, lo que ayuda a motivar a los alumnos (figura 2.24). No se suministra nada referido a su desarrollo, tales como librerías empleadas, SDK's, etc. pero utiliza el sistema de marcadores, tal como ARToolKitPlus.



**Figura 2.24:** SecondSight

## INVIZIMALS

El sistema Invizimals (INV10) es un videojuego desarrollado para la videoconsola portátil Sony PSP. Dentro de los dispositivos móviles, Invizimals representa una novedad ya que es la primera aplicación de RA para este tipo de dispositivos. Este juego se basa en la captura y recolección de criaturas fantásticas (figura 2.25), usando para ello unos marcadores diseñados para tal fin. Al igual que con otro tipo de sistemas, mediante los marcadores podemos visualizar las criaturas con un alto grado de orientación con respecto al usuario, para poder realizar batallas contra otro tipo de criaturas y jugadores. Una de las características más representativas es el alto grado de detalle de las criaturas, debido a las altas prestaciones que ofrece este dispositivo en

cuestión de gráficos 3D. Precisamente uno de los mayores problemas de la RA para dispositivos móviles es que la representación de los objetos 3D debe ser restrictiva en cuanto al número de polígonos que puede manejar la GPU, y esto no ocurre en PSP.

Invizimals pone de manifiesto que la RA es un campo de investigación que se abre paso a través de otro tipo de ámbitos meramente científicos o educativos, y da a conocer esta tecnología al gran público.



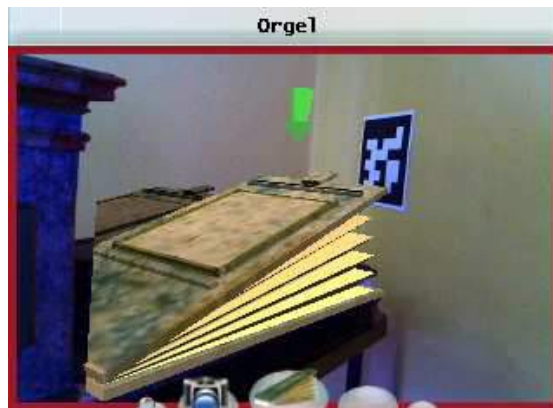
**Figura 2.25:** Invizimals, videojuego de RA

## MARQ

MARQ (Mobile Augmented Reality Quest) tiene como objetivo desarrollar una guía electrónica para los museos sobre la base de un dispositivo tipo PDA, que ofrece un entorno 3D totalmente interactivo de RA para los visitantes de dicho museo. El enfoque basado en PDA presenta un bajo coste económico por dispositivo y alta escalabilidad.

Los visitantes utilizan la PDA como guía personal que muestra una imagen 3D para combinar elementos reales y virtuales (ver figura 2.26). Así, las exhibiciones reales del museo son aumentadas por una parte virtual. La aplicación conoce la posición del usuario y proporciona información del contexto. La guía se convierte así en una especie de "lente mágica" capaz de

visualizar los objetos de otro modo ocultos. Al mirar a través de la pantalla, los visitantes pueden ver visualizaciones y animaciones superpuestas en las exposiciones reales. La guía muestra el contenido multimedia, pero también actúa como un asistente inteligente y puede llegar a substituir a una autoguía. También lleva consigo un sistema multi-usuario de comunicación para la interacción entre los grupos de visitantes.



**Figura 2.26:** Mobile Augmented Reality Quest

La aplicación se distribuye en forma de juego de equipo orientado y se ha utilizado con equipos de visitantes (de edad entre 12 y 16), que colocándose en el papel de investigadores, tratan de resolver una serie de puzzles para determinadas situaciones. Cuando un puzzle es resuelto, se proporcionan más pistas para continuar con el juego. Un aspecto interesante de este proyecto es que se muestra gran cantidad de información a los visitantes, haciendo de esta aplicación un sistema para el aprendizaje.

## 2.7 Sistemas sin marcadores

En el campo de la RA, la utilización de marcadores ha sido en muchas ocasiones el enfoque más clásico a fin de saber con exactitud la posición de la cámara. En esta parte, describiremos una nueva forma de representación de RA sin marcadores, donde el uso de dispositivos externos como el GPS y la brújula digital resultan indispensables con tal de que los objetos de la escena guarden una coherencia visual para quien esté visualizando la escena captada en la captura de vídeo. Este tipo de realidad aumentada se está empleando mayoritariamente en exteriores, y su campo de aplicación es la neogeografía.

Sin la utilización de marcadores, no sabremos cuál es el punto de vista del observador, haciendo que los objetos insertados no se adecuen al entorno o escena donde están situados. Esto hace que sin marcadores, difícilmente podamos visualizar correctamente objetos virtuales complejos con las técnicas que hay disponibles actualmente. Para suplir esta gran carencia, se están utilizando dispositivos externos que junto con un correcto tratamiento por software, nos permita ver los objetos de forma adecuada. Puntualizaremos, que en el estudio de este tipo de aplicaciones, no se han encontrado visualizaciones de objetos complejos, y la representación tridimensional no se contempla en casi ninguna de ellas. La forma más común de representación son en forma de etiquetas (2D) y gráficos vectoriales.

En los primeros capítulos se habló de que los dispositivos móviles cada vez están incluyendo más sensores y nuevo hardware, lo que hace que la experiencia entre el usuario y el dispositivo sea más enriquecedora. Dichos sensores son los que permiten que tengamos cada vez más, y sobre todo en el ámbito comercial, nuevas aplicaciones de RA, con las serias limitaciones de no disponer de una exactitud grande del punto de vista del observador. Aunque ya se habían implementado con este tipo de dispositivos/sensores, no es hasta la actualidad cuando se han empezado a convertir en sistema de realidad móvil, debido al tamaño reducido de los componentes hardware. Pasamos a

realizar una descripción de estos sensores y la forma en que son utilizados para lograr esa aproximación.

*GPS*: utilizado para conocer la localización del observador y situarse en el entorno. Permite conocer la posición a través de las coordenadas GPS, pero no la orientación.

*Brújula digital*: con este sensor se puede conocer hacia qué dirección estamos observando. Nos proporciona la dirección hacia donde estamos apuntando con la cámara, para así sólo mostrar aquellos puntos en forma de etiquetas que estén aproximadamente sobre la misma dirección.

*Acelerómetro*: una vez conocidas la posición y dirección, es recomendable conocer la orientación de la cámara (punto de vista del observador). Esto se realiza en base al funcionamiento de este sensor, que nos proporciona unos valores en X, Y, Z, que convenientemente tratados, nos permitirán conocer cuál es la orientación de la cámara con respecto a un plano vertical (el suelo se considera horizontal).

## 2.8 Librerías de desarrollo

### **ARtoolKitPlus**

Esta librería para RA es una variante de ARtoolKit, optimizada para dispositivos móviles. Fue desarrollada por la Universidad Tecnológica de Graz (TUG) (WAGNER07) y se liberó su código debido a la alta demanda de peticiones para su uso. Está destinada a calcular la orientación y la posición de la cámara relativa a los marcadores, todo ello en tiempo real. Debido en parte a que su desarrollo fue cerrado, no es una librería pensada para usuarios con poca experiencia en el desarrollo de aplicaciones de RA, puesto que no hay una gran documentación al respecto y tanto su desarrollo como su soporte se

encuentran parados desde 2007. Ha sido reemplazada por StudierStube Tracker y StudierStube ES, también desarrollado por la misma universidad.

En un capítulo posterior de este proyecto se explicarán los puntos más importantes que tiene esta librería para el desarrollo con dispositivos portátiles.

### **Nyartoolkit**

NyARToolKit es una librería de clases derivada de ARtoolKit, como ocurre con ARToolKitPlus. Su primera versión fue publicada en 2008, y escrita íntegramente en Java. Esto hace que sea más lenta en ejecución, pues Java está típicamente compilado en un bytecode (código intermedio más abstracto que el código máquina) para posteriormente ser interpretado por la máquina virtual de Java, pero le aporta la gran ventaja de ser multiplataforma e independiente de la arquitectura. No provee el acceso a la captura de vídeo, únicamente se encarga de analizar y extraer la posición y orientación de la cámara respecto a los marcadores y está optimizada para ser usada en dispositivos móviles (ver figura 2.27).

Ha sido reescrita también para otro tipo de lenguajes de programación, tales como C#, C++ e incluso para el sistema operativo Android, del cual se dispone de un conjunto de plugins para Eclipse, la principal herramienta de desarrollo para este sistema operativo desarrollado, entre otros, por Google Inc. Su uso está restringido para aplicaciones de uso no comercial bajo licencia GPL (GPLv3), siendo posible obtener licencias para desarrollo de aplicaciones comerciales si se desea.

A continuación se muestra una lista con las características a destacar de NyArtoolkit:

- Framework sencillo para el desarrollo de aplicaciones de RA.
- Soporte para librerías de vídeo y diversos formatos de imagen (RGB, YUV420P, YUV).
- Funciones de calibración de la cámara.
- Soporte con gran variedad de librerías gráficas, gracias a la multitud de lenguajes en la que está portada (OpenGL ES, VRML 3D, etc.).



- API modular y bien estructurada.
- Soporte para dispositivos Android.



**Figura 2.27:** Ejemplos de NyArToolkit

### **Metaio – Unifeye Mobile SDK**

Este software está desarrollado por la compañía Metaio. Unifeye Mobile es un SDK para desarrolladores disponible para sistemas operativos móviles más extendidos en la actualidad, como son Windows Mobile, Android, iPhone OS y Symbian. Ofrece un API para el reconocimiento de marcas ID y plantillas (figura 2.28). Como toda librería destinada a ser ejecutada en dispositivos portátiles, ha sido optimizada teniendo en cuenta las limitaciones hardware existentes hoy en día.

Veamos en la siguiente lista cuáles son las características más importantes:

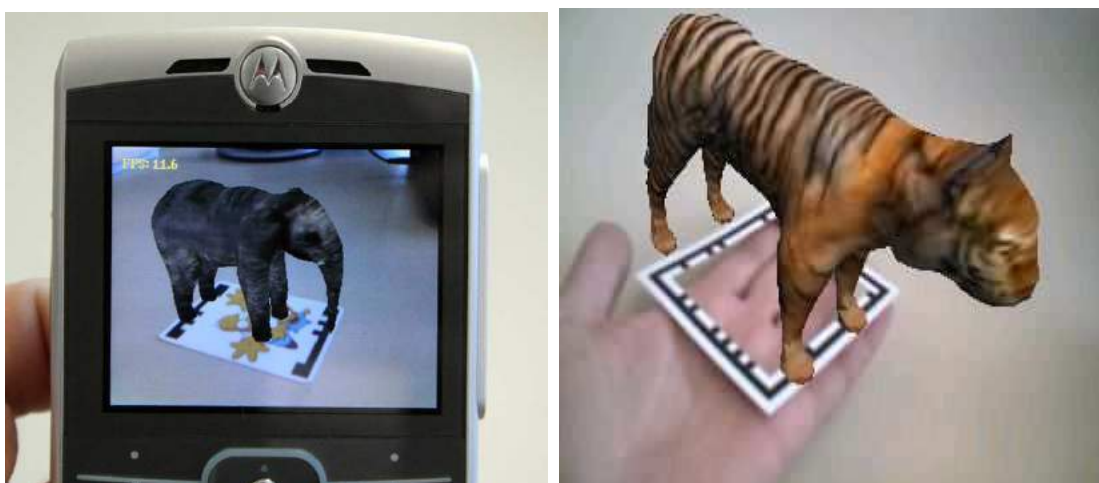
- Renderizado de animaciones 3D en tiempo real basado en OpenGL ES 1.X.
- Soporte para los desarrolladores mediante una API.
- Formatos 3D nativos: md2 y obj.
- Rastreo (tracking) basado en marcadores ID y plantillas.
- Soporte para el acceso a captura de vídeo.



**Figura 2.28:** Metaio – Unifeye Mobile SDK

### Studierstube ES

El framework StudierStube (y su extensión StudierStube ES) es una librería de visión por computador para la detección de la orientación y la posición de las marcas 2D con respecto a la cámara del dispositivos donde se esté ejecutando. La anterior versión es ARToolkitPlus, luego está desarrollada también por la Universidad Tecnológica de Graz (TUG). StudierStube ES ofrece características únicas y mejoras con respecto a su predecesor, siendo algunas de sus características que es multiplataforma y que soporta gran variedad de dispositivos que disponen de aceleración para gráficos 3D (figura 2.29). Pese a que guarda similitudes con ARToolkitPlus, éste ha sido reescrito para obtener un alto rendimiento, tanto en PC como en plataformas móviles.



**Figura 2.29:** Tracking con Studierstube

Si lo comparamos con ARToolkitPlus, StudierStube requiere hasta un 80% menos de memoria (unos 100 KByte aproximadamente) y es capaz de ejecutarse hasta dos veces más rápido. Es pues, el fruto de un trabajo de años de experiencia dentro del campo de RA en dispositivos móviles. Pasamos a enumerar las características de esta librería:

- Desarrollo cerrado (no se distribuye el código).
- Completa API basada en clases.
- Diferentes tipos de marcadores: plantilla, ID, DataMatrix, con marco, con marco partido y por puntos.
- Gran rendimiento en dispositivos con bajas prestaciones. Algunas pruebas realizadas han obtenido hasta 185 imágenes por segundo con procesadores de 312 Mhz.
- Mejor aprovechamiento de la memoria. El rastreo de una imagen de 320x240 en formato YUV12 requiere menos de 100KB, ya que depende de la resolución de la cámara y cada píxel ocupa un byte.
- Soporte multiplataforma: Windows XP, WindowsCE & Windows Mobile, Symbian, Linux, MacOS, Iphone.
- Disponibilidad de hasta 4096 marcadores ID.
- Formatos de píxel de la cámara: RGB24, RGB32, RGB565 y YUV12.
- Ancho del marco del marcador variable.
- Gran precisión de la posición/orientación, gracias al algoritmo "Robust Planar Pose".
- Soporte para la calibración de la cámara mediante el toolbox de MATLAB.
- Umbralización automática. Algoritmo de umbralización adaptativo para la corrección del binarizado en condiciones con luz excesiva.
- No hay límite en la resolución del vídeo.

Por otra parte, cabe destacar que StudierStube (ES) no da soporte para el acceso a la cámara, no tiene implementado en su API funciones para el renderizado, no detecta marcas naturales y no soporta hardware específico.

## Layar

El software Layar es formado por una API que provee una arquitectura software para el desarrollo de aplicaciones de RA sin la utilización de marcadores matriz 2D. Dichas aplicaciones, denominadas capas, son incorporadas al navegador móvil de RA de Layar (ver figura 2.30), el cuál es capaz de mostrar información de toda índole, muy enfocada al ámbito del entretenimiento, turismo e incluso a los negocios. Una muestra de ello es que podemos visualizar a través de su navegador las localizaciones más cercanas a nuestra posición, como museos, restaurantes, etc. Para ello, utiliza los sensores incorporados en los nuevos dispositivos móviles (ver apartado 2.7).

De cara al desarrollador, Layar está compuesto de varios sistemas encargados de proveer la información. Así nos encontramos con la aplicación cliente que se ejecuta en el dispositivo, un servidor central donde se accede para obtener los datos y otro servidor privado para el desarrollador donde gestionar los datos, que son enviados al servidor central y finalmente visualizados en la aplicación a través de la cámara incorporada. Layar también nos brinda la posibilidad de mostrar objetos 3D, utilizando un formato de representación de objetos 3D optimizado para móviles.

Layar está disponible para su descarga gratuita, y los desarrolladores interesados pueden hacer uso de la API para crear sus propias capas. Actualmente, únicamente está disponible para el iPhone 3GS y algunos dispositivos de gama alta con el sistema operativo Android incorporado.



**Figura 2.30:** Layar en funcionamiento

# Capítulo 3

## Implementación del proyecto

### 3.1 Introducción

En este capítulo se presentará la funcionalidad de la aplicación diseñada y como se ha llevado a cabo su implementación, explicando las herramientas software utilizadas y detallando los aspectos más relevantes que han tenido lugar en la elaboración de esta aplicación.

Primeramente, realizaremos una descripción del sistema desarrollado para PC. Se hablará de cuáles eran sus objetivos a conseguir, un breve resumen de su implementación y algunos aspectos a tener en cuenta para la realización de este trabajo. No obstante, si se quiere tener una visión más completa sobre él, hará falta una lectura detallada del trabajo (CARRION04).

En segundo lugar, se hablará sobre el dispositivo móvil utilizado en este trabajo, analizando su hardware y qué opciones nos brinda de cara a desarrollar aplicaciones de RA, entre otras cosas. Para finalizar este capítulo, explicaremos todo el conjunto de herramientas software empleadas y cómo se han utilizado para implementar el proyecto.

En este último capítulo dedicado al desarrollo del proyecto, nos centraremos en cómo está realizado el programa. Se verá su funcionamiento, su estructuración y todos aquellos elementos que sean necesarios explicar para comprender cómo está elaborado el programa. Además, se realizará un análisis Bottom-up a fin de resumir los pasos necesarios para realizar aplicaciones de RA en el dispositivo escogido, aunque esto puede extrapolarse al resto de dispositivos sin muchos cambios significativos.

## 3.2 Descripción del anterior proyecto

En este apartado trataremos de explicar cómo está estructurada la aplicación existente (CARRION04) de la cual se basa este proyecto. Para empezar vamos a definir el propósito de la aplicación ya desarrollada, pues ésta no tiene como única finalidad la creación de una aplicación RA, sino que pretende ser una herramienta de ayuda para tratar alteraciones psicológicas, como la provocada por insectos pequeños y que se deriva en una fobia hacia ellos. En este caso se trabajará con un tipo de insectos, las cucarachas.

La terapia con RA posee unas características que la convierten en una tecnología adecuada para generar condiciones que mejoran determinados procedimientos de tratamiento psicológico. Se puede deducir que con la RA, el escenario generado podrá tener un impacto más fuerte sobre el paciente, que le permitirá percibir los objetos generados como si formasen parte del escenario real donde se encuentra. Dicho sistema proporcionará al paciente un procedimiento de tratamiento psicológico para su fobia y al psicólogo/a que lo trate, una herramienta terapéutica que le ayude en su labor y le permita llevar un control y seguimiento total de la terapia. Se definieron unos requisitos de implementación para lograr una buena aplicación para el tratamiento de esta fobia, son los siguientes:

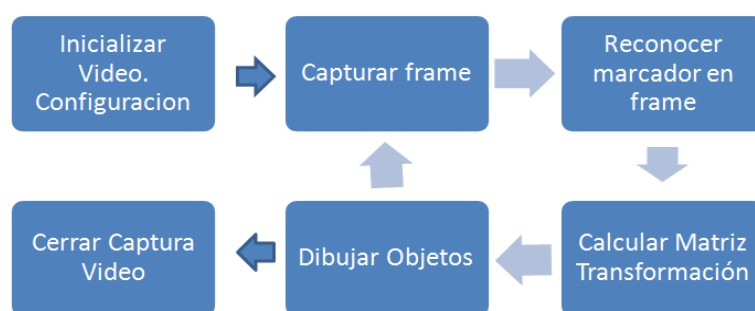
1. Debía ser un sistema de tiempo real, es decir, debía permitir al usuario o al psicólogo interactuar y manejar la aplicación en todo momento durante la exposición sin retrasos visuales aparentes.
2. El realismo era un objetivo muy a tener en cuenta en la aplicación. El nivel de realismo gráfico en los animales debían ser de gran calidad, así como su movimiento y animación, para conseguir que la aplicación fuera creíble y tuviese el efecto deseado.
3. El sistema debía permitir que, durante el tratamiento, se planteasen al usuario una serie de estímulos relacionados con el

problema a tratar. Estos estímulos, la forma representarlos y su grado de intensidad deberían estar a la altura del nivel de eficacia perseguido.

4. El sistema debía desarrollar situaciones con las que el paciente se pudiese encontrar en la vida real. Por tanto, se incluyeron objetos que una persona usaría en una situación real para interactuar con la aplicación, como son: una paleta matamoscas, un insecticida o un recogedor para coger la cucaracha una vez muerta.

Pasemos ahora a describir la implementación del proyecto del cual se basa este trabajo. Se utilizó la librería de RA ARToolkit, la cual se ha utilizado en otros trabajos como (WOODS03) y (BILLINGHURST01). Los detalles y funcionalidades de esta librería se verán reflejados en el capítulo correspondiente a la librería ARToolkitPlus, aunque si bien ésta está optimizada para dispositivos portátiles, comparte las mismas características principales, como son la obtención de la matriz de proyección a fin de obtener el punto de vista del observador, entre otros.

Como toda aplicación de RA que utiliza ARToolkit, consta de un bucle principal de ejecución englobado dentro una serie de pasos generales, tal como se muestra en la figura 3.1.



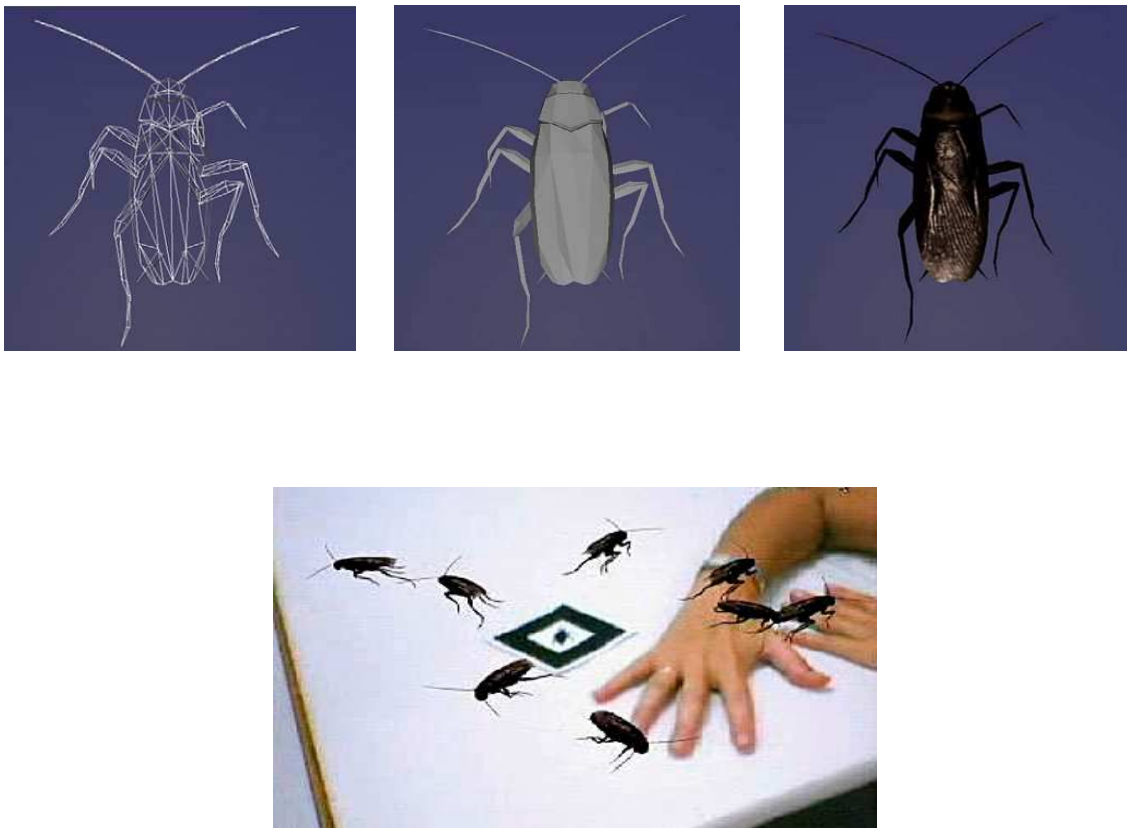
**Figura 3.1:** Bucle de ejecución de RA

Este bucle de ejecución no difiere demasiado de la implementación desarrollada para el actual proyecto, pues ambos están basados en RA mediante marcadores. Las funciones que realizan cada bloque del diagrama pueden consultarse en (CARRION04) en las secciones correspondientes.

### Capítulo 3: Implementación del proyecto

Tal como se ha descrito en el punto 2 de los requisitos de implementación a conseguir, el realismo gráfico ha de ser alto, por los motivos que se explicaron en ese punto. Para ello, el autor modeló mediante un software CAD (3D Studio Max), el modelo 3D de las cucarachas, generando dos modelos de cucaracha: estática y muerta.

Otro aspecto que se tuvo en cuenta fue la creación de una textura para simular el aspecto y el color del insecto y así dotarle del realismo necesario. Ya por último, se utilizó VRML para crear y simular el movimiento de las patas. En las figura 3.2 podemos ver el resultado final del modelado, junto con su modelo alámbrico y sólido, así como una imagen de la aplicación.



**Figura 3.2:** Modelado de las cucarachas y ejecución de la aplicación



El último punto a resumir sobre el anterior trabajo son las funcionalidades requeridas en el punto tres de los requisitos de implementación, que es el referido a los estímulos que pueda recibir el paciente por parte de la aplicación, y para ello se realizaron unas funcionalidades básicas para el tratamiento de este tipo de fobias, adaptadas al tipo de insecto que se estaba tratando. Las opciones de las que dispone son:

- Inserción de objetos. Permite insertar en la escena un número definido de cucarachas. Se define un máximo de objetos para visualizar a fin de no ralentizar la aplicación.
- Escalado de objetos, para agrandar o reducir dichos objetos.
- Movimiento. Los objetos se desplazan por la escena, teniendo en cuenta unos límites de desplazamiento, el cual los objetos realizan un giro de 180° y continúan trasladándose en dirección contraria.
- Detener el movimiento. Detiene los objetos en la posición actual.
- Iniciar. Borra los objetos de la escena y coloca un objeto (cucaracha estática) en el centro.
- Matar. Se carga en la escena el objeto de cucaracha muerta, eliminando el de cucaracha estática (incluido si se están moviendo).
- Espray. Reproducción de un sonido de espray y eliminación de objetos tal como se hace en el anterior punto.
- Coger. Carga una cucaracha muerta para su manipulación junto con el marcador.

Algunas de estas funcionalidades serán implementadas en el sistema que se va a desarrollar en este proyecto. Se intentará que éstas sean lo más parecidas posibles, ya que se desea la misma funcionalidad, aunque con las restricciones que tiene un dispositivo móvil. Como se comentó, la finalidad de este trabajo es que la aplicación sea efectiva para el tratamiento de fobias a animales pequeños. Para comprobar dicho objetivo se realizó el tratamiento a unos pacientes reales (JUAN05).

Antes se han descrito cuatro objetivos claros y bien definidos (exigibles) a fin de tener mantener un alto nivel de realismo en la escena. Partiremos de esos objetivos matizando en cada uno de ellos qué problemas puede haber si estamos realizando su implementación en un dispositivo portátil.

El punto número uno habla sobre la interacción entre el usuario y la aplicación, y de cómo ésta debe responder adecuadamente para no producir retrasos visuales aparentes. Para conseguir ese propósito, se intentará ser lo más fiel posible al diseño original, pero sin que ello repercuta en el rendimiento, pues entonces no estaríamos cumpliendo el objetivo. Los posibles cuellos de botella con los que nos podemos encontrar son: cálculo de la matriz de proyección en base a marcadores, acceso a captura de vídeo y representación de objetos tridimensionales.

El segundo punto habla del realismo gráfico de los objetos virtuales mostrados en pantalla. Éste quizás es un aspecto crítico, puesto que un dispositivo móvil no tiene hoy en día la misma capacidad que un equipo PC en cuestión de representación de gráficos 3D, tanto en las limitaciones hardware como software, que dependen de la primera. Se intentará buscar un compromiso entre vistosidad y rendimiento.

El tercer punto trata sobre los estímulos a los que el usuario debe someterse durante el tratamiento. En base a los puntos uno y dos y con las limitaciones del dispositivo, se desarrollará una vez más, una aproximación lo más fiel posible al proyecto base.

### 3.3 Diseño de la aplicación

El presente proyecto es una aplicación de RA para el dispositivo iPhone 3GS basada en un proyecto anterior que pretendía ayudar al tratamiento de fobias/ansiedad frente a animales pequeños, tales como cucarachas. Asimismo, también se puede comprobar el avance del estado de la RA en los dispositivos móviles, un hecho que hace tan solo unos cuantos años era impensable debido a las limitaciones de hardware que presentaban los dispositivos.

Se pretende por un lado, realizar una implementación con un diseño lo más parecido al anterior trabajo, con el fin de poder comprobar las bondades de la RA en un Smartphone como el iPhone. Es por ello que se han mantenido muchas de la funcionalidades que tenía el sistema original, tales como movimiento de las cucarachas, acción de spray, detener el movimiento, etc. El sistema se inicializará y a continuación se podrán añadir insectos a la escena, hasta un máximo permitido. También será posible quitar dichos insectos, sin que esto afecte al resto de insectos de la escena. Con la acción de spray, se oirá un sonido parecido cuando se utiliza un bote de spray, a modo de simulación, para eliminar (matar) a un gran número de cucarachas. También, mediante un control deslizante, se podrá cambiar el tamaño de los insectos, fijando un mismo tamaño para todos por igual. La última de las opciones consistirá en mostrar una cucaracha muerta en el marcador para poder interactuar con el insecto. Esta acción pretendía en el proyecto anterior, desde un punto de vista terapéutico, hacer que el paciente pierda el miedo a este tipo de insectos una vez que ya estén muertos.

#### 3.2.1 Dispositivo móvil: iPhone

Al inicio de este proyecto nos planteábamos cuáles eran las limitaciones actuales que tienen los dispositivos móviles. Esta situación se plantea como un reto a la hora de poder realizar una portabilidad con éxito del proyecto base, y por ello se hace necesario encontrar aquel dispositivo que nos brinde una

serie de ventajas y facilidades a la hora de programar. En nuestro caso particular, se ha escogido un iPhone 3GS. A fecha de realización de este proyecto, es el dispositivo que ofrece mejores herramientas para nuestro propósito y aparte es uno de los Smartphone más versátiles y potentes que existen.

El iPhone (ver figura 3.3) es un teléfono inteligente (Smartphone) de la compañía Apple Inc. Dispone de una pantalla táctil de 320x480 píxeles, cámara de vídeo de dos megapíxeles y capacidad para reproducir aplicaciones. Existen dos modelos, 3G y 3GS. Este último, es hasta dos veces más rápido en tareas como la ejecución de aplicaciones, además de dar mayor soporte al estándar OpenGL ES 2.0 para ofrecer mejores gráficos 3D y su cámara es de 3 megapíxeles, AGPS, acelerómetro y brújula digital. Por lo tanto, es un dispositivo aceptado para desarrollar aplicaciones de RA (ver figura 3.4).



**Figura 3.3:** Smartphone iPhone

La especificación del iPhone puede verse a continuación (sólo se muestran la especificación que puede ser interesante para el desarrollo del proyecto):

Tamaño de la pantalla: 3.5 (89 mm) 320 x 480 píxeles en 163 ppi, Multitouch,
Cámara de 2 MP, 1600x1200 píxeles (3 MP en el modelo 3GS)

Samsung S5L8900 (620 MHz, procesador ARM 1176, coprocesador de 3D de gráficos PowerVR MBX)
Memoria: 128 MB DRAM
Sistema Operativo: iPhone OS



**Figura 3.4:** RA en iPhone

## 3.2.2 Herramientas software

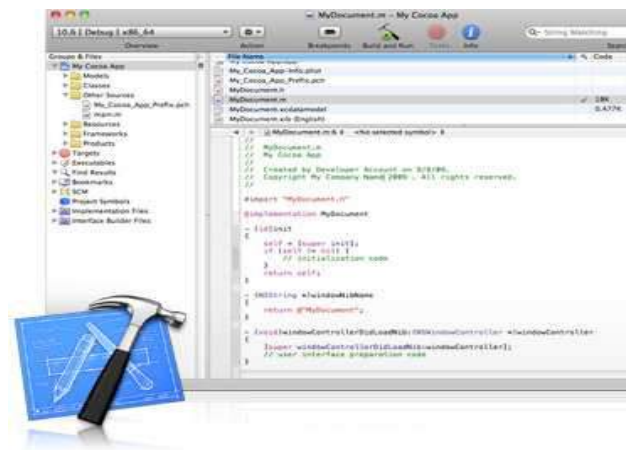
### 3.2.2.1 iPhone SDK

El iPhone dispone de un SDK de desarrollo que se distribuye gratuitamente para todos aquellos desarrolladores que quieran programar con su API. Esta API nos ofrece un conjunto amplio de librerías/frameworks para crear potentes y dinámicas aplicaciones para el iPhone (es posible desarrollar también para iPod Touch y iPad). La versión utilizada para este proyecto ha sido la SDK 3.1.3, lanzada a principios de febrero de 2010. Junto con la SDK, se distribuye la herramienta de desarrollo, IDE Xcode.

Xcode es el entorno de desarrollo integrado (IDE) de Apple Inc. Y al igual que la SDK, se distribuye libremente para la comunidad de desarrolladores. Incluye otro tipo de herramientas, tal como Interface Builder e Instruments. Desarrollar con Xcode es igual que con cualquier otro IDE, pues posee más o menos las mismas funcionalidades que presentan estas

### Capítulo 3: Implementación del proyecto

herramientas. Permite la depuración de código mediante una interfaz gráfica, posee autocompletado de código (respetando la sintaxis estándar del lenguaje que se utilice), permite instalar aplicaciones en el dispositivo con tan sólo modificar algunos parámetros, gestionar los dispositivos conectados, compilar junto con los certificados de desarrollador necesarios y también es posible depurar las aplicaciones directamente sobre el iPhone.



**Figura 3.5:** Entorno de programación Xcode

Tal como puede verse en la figura 3.5, la ventana de proyecto Xcode está estructurada en tres partes bien diferenciadas. La barra superior sirve para buscar referencias dentro de nuestro código, crear puntos de interrupción, compilar el código y elegir el tipo de dispositivo en el cual vamos a depurar el código, el simulador o el dispositivo físico. El panel central es el del editor, donde podremos escribir código, editarlo, etc. La parte izquierda, llamada *Group & Files*, contiene la jerarquía de archivos y algunas opciones de propiedades del proyecto. Hay cinco subcarpetas dentro del panel, y son:

- *Classes*: donde se guardan la mayor parte del código generado. Permite crear subcarpetas para poder organizar nuestros archivos de código de una manera organizada.
- *Other sources*: contiene ficheros que no son generados explícitamente por el desarrollador. Incluye el método `main ()` de la aplicación así como un fichero donde se encuentran todos los ficheros de cabecera

precompilados de los frameworks usados en la aplicación. Esto hace que la fase de compilación sea más rápida cada vez que compilamos.

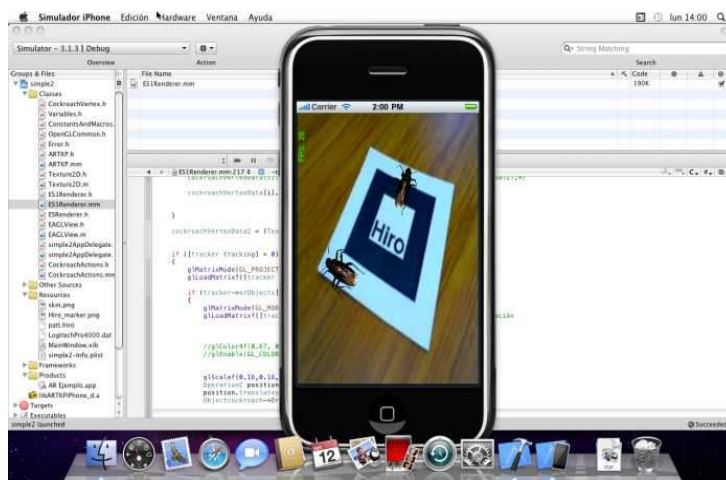
- *Resources*: contiene los ficheros que son incluidos en la aplicación pero no son de código fuente. Puede contener imágenes, iconos, archivos de música y cualquier otro fichero que se necesite en la aplicación.
- *Frameworks*: contiene las librerías que se usan en la aplicación. Por defecto se incluyen las más básicas, aunque es posible añadir nuevas posteriormente.
- *Products*: esta subcarpeta contiene la aplicación una vez se ha compilado el proyecto. La extensión de nuestras aplicaciones compiladas es .app.

Interface Builder es la herramienta que incluye Xcode para la generación de prototipos de interfaz gráfica. Permite crear potentes interfaces sin necesidad de escribir ningún código. Es posible añadir nuevas ventanas, botones, listas y demás controles que serán indispensables en la creación de aplicaciones para iPhone. La forma de construir este tipo de interfaces sigue el esquema Modelo-Vista-Controlador. Los objetos agregados están conectados dinámicamente a su código nativo en tiempo de ejecución. Por eso los cambios en la interfaz no necesitan volver a compilar el código, y los cambios en el código no necesitan volver a compilar la interfaz. Para el diseño de la interfaz se ha utilizado, como es obvio, Interface Builder para incluir todos los controles que manejan las funcionalidades.

El entorno Xcode dispone también de un completo simulador para testear las aplicaciones (figura 3.6) sin necesidad de disponer del dispositivo real. Es el banco de pruebas perfecto para asegurarse de que la aplicación que se está desarrollando puede funcionar en el iPhone. Con esta herramienta se ahorra mucho tiempo en la fase de testeo, y es por ello que resulta esencial en la fase de programación. Durante este proyecto, gran parte del desarrollo se ha realizado con el simulador, por ello los ejemplos que se muestren serán con la interfaz del simulador. Como se puede intuir, no es posible acceder a la captura de vídeo desde el simulador, así que como más adelante se explicará, se ha utilizado una textura con un marcador para que ARToolKitPlus pueda

## Capítulo 3: Implementación del proyecto

obtener los datos de la matriz de transformación a través de los datos de la textura.



**Figura 3.6:** Simulador integrado en Xcode

La herramienta Instruments, también disponible en la SDK, es un conjunto de aplicaciones que tienen como finalidad la ayuda al programador en tareas como la depuración de código, la revisión de la cantidad de memoria utilizada, gestión de ficheros, visualizar la ejecución de nuestro programa, localización de cuellos de botella, etc. Instruments recoge datos tales como discos, memoria, o el uso de CPU en tiempo real, incluso desde un iPhone conectado. Los datos recogidos se representan gráficamente como pistas en el tiempo, por lo que es fácil de identificar áreas problemáticas de código.

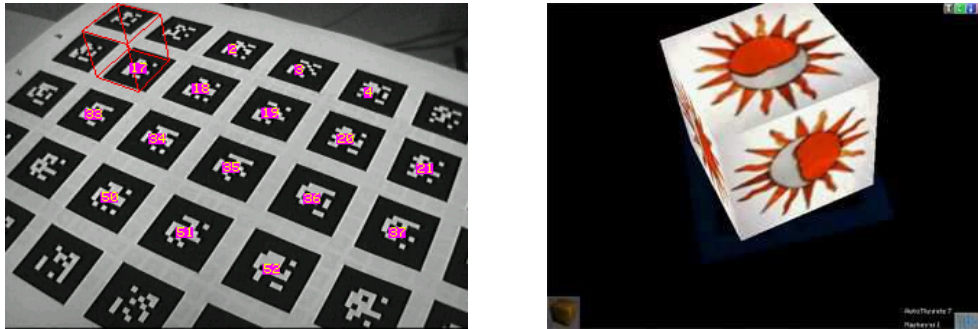
### 3.2.2.2 ARToolkitPlus

Uno de los aspectos más críticos a la hora de desarrollar sistemas de RA en dispositivos móviles es la utilización de una buena librería que nos proporciona la matriz de transformación de los objetos de la escena. En este proyecto, se ha utilizado la librería ARToolkitPlus (ARPLUS07). Fue ideada a partir de otra librería, ARToolkit, con la característica principal de que iba destinada para dispositivos portátiles, que disponen de un hardware técnicamente inferior.



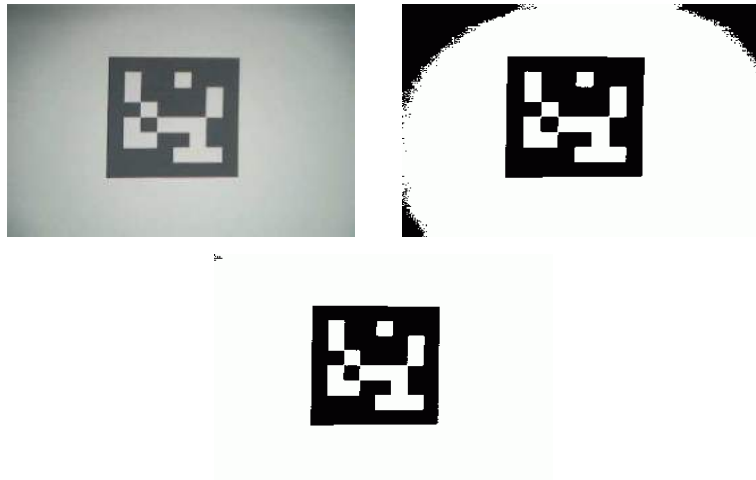
Las características más importantes añadidas con respecto a ARToolKit son las siguientes:

- API basada en clases. Se crea una clase llamada Tracker, y otras 2 derivadas, una para el reconocimiento de marcas individuales, y la otra para reconocimiento de múltiples marcas. Esto facilita el desarrollo y mejora la gestión de memoria, permitiendo utilizar más de una instancia de ARToolKit por proceso. Además, se sustituyen las configuraciones de preproceso a bajo nivel (#define...), por plantillas de parámetros.
- Soporte para imágenes de escala de grises de 8 bits. En sistemas empotrados, el acceso a memoria es muy costoso, por lo que el uso de imágenes con un bajo bitrate mejora la velocidad del sistema.
- Soporte para imágenes de RGB565 de 16 bits. En sistemas empotrados como las PDAs, el formato de imagen que devuelven es RGB565, por lo que permitir el uso del formato nativo (ayudándose de LUTs), elimina la necesidad de convertir a otros formatos de imagen.
- Implementación parcial del punto-fijo para PocketPC. Mejora la velocidad del sistema.
- Ancho del borde de marca variable, que permite que se dedique más porción de la marca al patrón, aumentando la probabilidad de detectar marcas muy pequeñas.
- Optimización de Thomas Pintaric en el indexado de punteros, que ocasiona una aceleración sobre el 10% en PCs (menor en PDAs).
- Utilización de una LUT (lookup-table) para evitar la distorsión de la lente.
- Un nuevo algoritmo para la estimación de la posición que deriva en un reconocimiento de marcas más estable.
- Implementa el algoritmo "Robust Planar Pose".
- Codificación simple de marcas. Se da la posibilidad de utilizar un reconocimiento de marcas con una ID codificada en lugar de las imágenes plantilla facilitadas. Esto permite utilizar hasta 512 marcas sin necesidad de entrenamiento, ni ralentizaciones (ver figura 3.7). Consiste en codificar un número de 9 bits, repetido 4 veces introducido en una matriz de 6x6.



**Figura 3.7:** Ejemplos de ARToolkitPlus

- Codificación BCH de marcas. Disponible desde la versión 2.0, es más robusto que la codificación explicada en el punto anterior, debido al uso de un algoritmo avanzado de CRC, aumentando el número de marcas hasta 4096 cuando se utiliza la codificación BCH.
- Soporte para coma flotante de simple precisión. Los sistemas empotrados no suelen soportar cálculos en coma flotante.
- Mejora de detección de patrones en modo half-res. Modifican `arGetPatt ()`.
- Umbralizado automático. ARToolkitPlus es capaz de analizar el patrón dentro de la marca y encontrar el píxel más oscuro y más claro, calculando la media y realizando un umbralizado dinámico. Si no ha encontrado una marca, entonces este valor se calcula de forma aleatoria.
- Gestión de memoria manual. Se permite gestionar de forma manual la reserva de memoria.
- Compensación del viñeteado. Algunas cámaras tienen pérdida de luz radial, esto es, que la imagen devuelta es más oscura a medida que se aleja del centro. Tras realizar el umbralizado, puede que devuelva una imagen con un efecto llamado viñeteado, que detecta erróneamente píxeles no válidos como elementos de interés (ver figura 3.8).
- Mejora en el modelo de calibración de cámara (herramientas de calibración de cámara para MATLAB).



**Figura 3.8:** Reducción del viñeteado en ARToolKitPlus

### 3.2.2.3 Blender

Blender es un software multiplataforma dedicado al modelado, animación y creación de gráficos tridimensionales. Se enmarca dentro del tipo de aplicaciones denominadas CAD (Computer Aided Design).

Se puede utilizar para modelar, aplicar texturas, simulaciones de agua, animación, renderizado, partículas y otras simulaciones, edición no lineal, composición, y la creación de aplicaciones 3D interactivas, incluyendo juegos. El programa fue inicialmente distribuido de forma gratuita pero sin el código fuente, con un manual disponible para la venta, aunque posteriormente pasó a ser software libre. Actualmente es compatible con todas las versiones de Windows, Mac OS X, Linux, Solaris, FreeBSD e IRIX.

Tiene una interfaz gráfica de usuario diferente, quizás poco intuitiva, pues no se basa en el sistema clásico de ventanas; pero tiene a su vez ventajas importantes sobre éstas, como la configuración personalizada de la distribución de los menús y vistas de cámara.

A continuación enumeramos las características más destacadas que presenta este programa:

### Capítulo 3: Implementación del proyecto

- Multiplataforma, libre, gratuito. El espacio en disco que ocupa este software es realmente pequeño comparado con otros paquetes de 3D, dependiendo del sistema operativo en el que se ejecuta.
- Capacidad para una gran variedad de primitivas geométricas, incluyendo curvas, mallas poligonales, vacíos, NURBS, curvas Bezier, etc.
- Junto a las herramientas de animación se incluyen cinemática inversa, deformaciones por armadura o cuadrícula, vértices de carga y partículas estáticas y dinámicas.
- Edición de audio y sincronización de vídeo.
- Características interactivas para juegos como detección de colisiones, recreaciones dinámicas y lógica.
- Posibilidades de renderizado interno versátil e integración externa con potentes trazadores de rayos libres como kerkythea, YafRay o Yafrid.
- Lenguaje Python para automatizar o controlar varias tareas. Es posible la creación de scripts para tratar los objetos poligonales que hay en la escena.
- Blender acepta formatos gráficos como TGA, JPG, Iris, SGI, o TIFF. También puede leer ficheros Inventor.
- Motor de juegos 3D integrado, con un sistema de bloques lógicos. Para más control se usa programación en lenguaje Python.
- Simulaciones dinámicas para partículas y fluidos.
- Modificadores apilables, para la aplicación de transformación no destructiva sobre mallas.
- Sistema de partículas estáticas para simular cabellos y pelajes, al que se han agregado nuevas propiedades entre las opciones de *shaders* para lograr texturas realistas.

Para nuestro proyecto, se ha utilizado Blender para importar los objetos 3D del anterior proyecto (cucarachas). Estos objetos 3D tienen el formato de fichero VRML, y es necesario importar estos datos y pasarlos a otro formato que

pueda manejar nuestra aplicación en iPhone. Estas acciones se verán explicadas en las siguientes secciones.

### 3.2.2.4 OpenGL ES

OpenGL ES fue diseñado por el grupo Khronos y es una API ligera para gráficos 3D utilizada en dispositivos móviles. Se trata de una potente API de bajo nivel que es muy similar a la de OpenGL y proporciona la mayor parte de la funcionalidad de OpenGL. El consorcio Khronos aprovechó la oportunidad para limpiar la API OpenGL, y resolver muchas cuestiones pendientes que requerían una remodelación. Está disponible en las plataformas más importantes de dispositivos móviles.

Una decisión de diseño importante para OpenGL ES es que se ha eliminado la redundancia de la API OpenGL. En consecuencia no hay glBegin / glEnd ya que a los programadores se les recomienda utilizar una matriz de vértices o estructuras de datos adecuadas, ya que son más eficientes. No hay compatibilidad con tipos primitivos que no sean puntos, líneas o triángulos. Por lo tanto, OpenGL ES no es compatible con quads o polígonos de n lados. Estas y otras reducciones en la API de OpenGL, han producido una API de gráficos compacta y más fácil de implementar. Como consecuencia del conjunto de características reducidas, OpenGL ES 1.x pueden ocupar unos pocos cientos de KB en un dispositivo portátil.

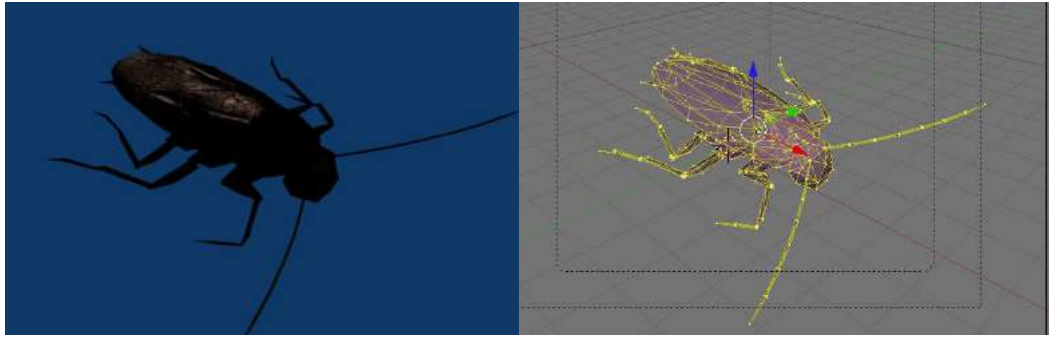
	OpenGL	OpenGL ES
<b>glBegin/glEnd</b>	sí	no
<b>Tipos de primitivas</b>	Todos	No hay Quads ni polígonos
<b>Tipos de datos</b>	double, int, float	float, int
<b>glDraw/Read pixels</b>	sí	glRead sólo
<b>Texturas</b>	1D,2D,3D,cubo	2D

### 3.3.3 Elementos virtuales

En una sección anterior ya se ha mostrado cuales son los objetos tridimensionales que van a ser utilizados en nuestro sistema de RA, que en nuestro caso van a ser cucarachas.

Para el diseño de estos insectos se ha partido del diseño previo realizado en VRML y que ya se utilizó en la aplicación para PC. Estos objetos disponen de demasiados polígonos como para que se puedan representar en iPhone y obtener buenos resultados de rendimiento, por ello se ha optado por reducir el número vértices de la malla poligonal. Los objetos originales tienen alrededor de 2300 vértices por cada uno de ellos y están representados por una malla poligonal formada por triángulos. Para reducir este número, se ha utilizado la herramienta Blender, para importar los archivos VRML y trabajar con ellos. Blender dispone de una característica muy importante, como es que puede trabajar con scripts en Python para manipular los objetos de la escena. En nuestro caso, se ha utilizado el siguiente script disponible en (IDEV10) para extraer los vértices de la malla, que juntamente utilizado con un estructura de datos adecuada para almacenar los vértices, hace posible su dibujo mediante OpenGL ES.

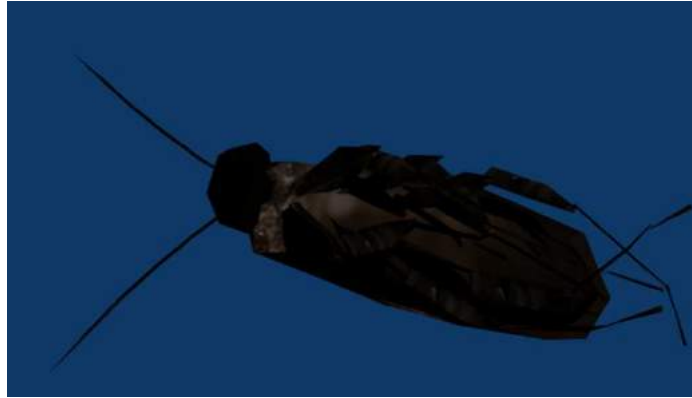
Previo paso a la ejecución del script, es necesario como se ha comentado, reducir el número de vértices. Esto se ha conseguido aplicando otro script disponible en el propio Blender, llamado "Poly Reducer", y tiene el efecto de eliminar polígonos manteniendo la forma del objeto, aunque si se ejecuta demasiadas veces dicho proceso, es posible que la forma del objeto difiera demasiado del objeto original, y eso en nuestro caso concreto, no es un objetivo nada deseable para que no se pierda realismo. En la figura 3.9 se puede ver un objeto, la cucaracha, durante el proceso de reducción de polígonos y su renderizado, tal como se mostrará en la aplicación.



**Figura 3.9:** Reducción de vértices con Blender

Con la reducción de polígonos se ha conseguido eliminar hasta casi la mitad de los vértices del objeto, quedándose en alrededor de 1300 vértices, una cantidad moderada y que el chip gráfico del iPhone puede manejar sin ningún problema. Una función de OpenGL ES, es la encargada de dibujar los vértices que han sido previamente exportados con el formato adecuado. Con este método, se han creado dos objetos de tipo cucaracha, para utilizarlas en la parte de movimiento, ya que se alterna entre uno y otro objeto mientras se están moviendo por la escena para dar una sensación de realismo, que es uno de los objetivos más importantes. Al igual que ocurre con la aplicación para PC, se ha añadido una textura al objeto para dotarlo de realismo, tal como se ha podido ver en el renderizado.

La figura 3.10 es la que representa la cucaracha muerta y se ha diseñado en base a la cucaracha normal. No se ha optado por utilizar la del diseño original ya que está tiene demasiados polígonos, más que la cucaracha original, y con aplicar el proceso de reducción de polígonos no se ha conseguido el resultado esperado, porque se pierde parte del realismo exigido en la aplicación. Con el nuevo diseño se han obtenido un número de vértices parecido a los de la cucaracha viva, luego es un resultado satisfactorio y se ha dado por válido para utilizar en la aplicación. Para este modelo se ha diseñado una nueva textura, puesto que la anterior sólo es válida para la parte superior del insecto, y no para el dorso, que es la parte más visible en este caso. El resultado se puede ver en la figura 3.10:



**Figura 3.10:** Modelado de cucaracha muerta

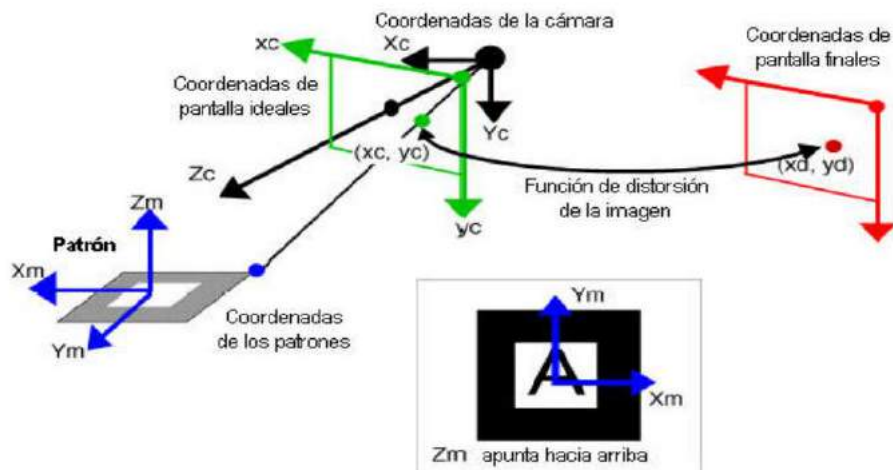
### 3.3.4 Funcionamiento de ARToolKitPlus

Una vez creados los objetos que se representarán en la escena, otro de los pasos más importantes es la puesta en marcha del sistema para la captura de vídeo y la utilización de ARToolKitPlus. En esta sección explicaremos cuáles han sido los pasos necesarios para poner en funcionamiento el sistema de RA sobre iPhone.

#### ***Cálculo de la matriz de transformación***

Ésta es la labor principal que realiza la librería ARToolKitPlus, y por extensión toda librería que utilice marcadores. Debe ser un proceso rápido ya que se debe superponer los objetos 3D mediante la aplicación de la matriz de transformación de la vista (punto de vista del observador) a fin de visualizarlos correctamente, y todo ello además con la sensación de que ocurre en tiempo real, o al menos, sin retrasos visuales aparentes para el usuario de la aplicación. En la figura 3.11 se muestra cómo es el proceso de obtención de la matriz de transformación de la vista, y en los siguientes párrafos se explicará en detalle dichos pasos.

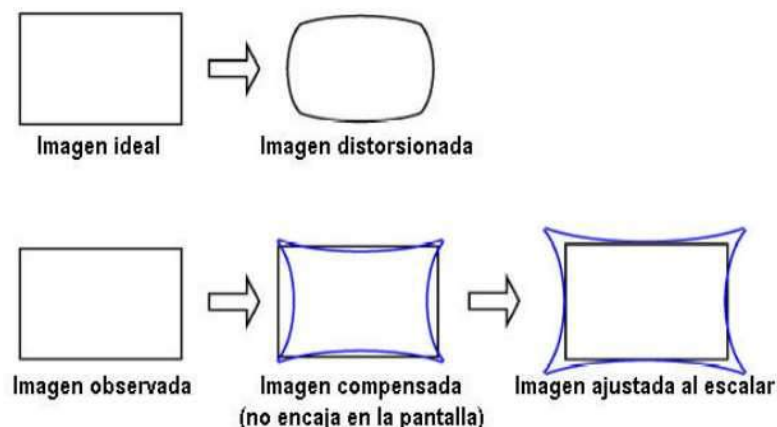




**Figura 3.11:** Cálculo de la matriz e transformación

En la figura 3.11 se pueden observar los principales sistemas de coordenadas implicados en el proceso. El posterior tratamiento y realización de los cálculos es parecido a los sistemas de visión por computador utilizados en gran multitud de campos, como la robótica, procesos industriales, etc.

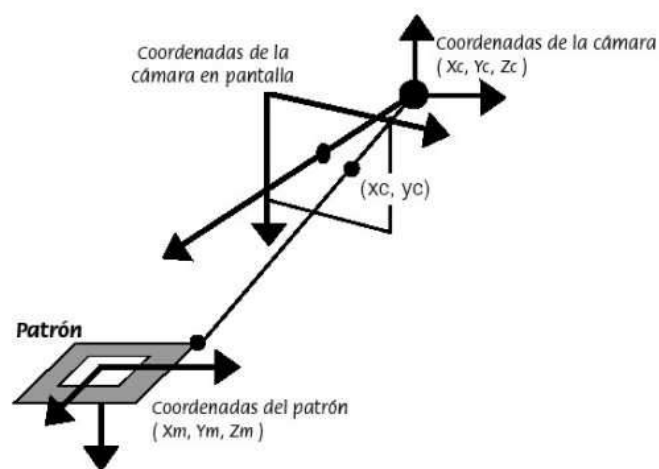
Al estar utilizando una cámara, se debe tener en cuenta el factor de distorsión producido por la lente de ésta y corregirlo (ver figura 3.12). Para nuestra aplicación, se ha utilizado un fichero de calibración de la cámara que se utilizó en otros proyectos, y este paso no ha sido necesario realizarlo. Mediante esta calibración de la cámara se realizan las operaciones para la corrección de dicha distorsión y obtener las coordenadas de la matriz para la cámara virtual, consiguiendo así dibujar los objetos correctamente.



**Figura 3.12:** Corrección de la distorsión

En nuestro caso particular la cámara se encuentra integrada en el dispositivo, y se produce este efecto de distorsión, corregido con la calibración de la cámara. No ha sido necesaria una nueva calibración, ya que los resultados finales obtenidos han sido satisfactorios.

Como base del espacio de coordenadas en el que se representan los monitores virtuales se utilizan marcas cuadradas de tamaño conocido. Las matrices de transformación desde estas coordenadas de las marcas a las coordenadas de la cámara ( $T_{cm}$ ) representadas en la figura 3.13 se realizan mediante un análisis de la imagen.



$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} V_{11} & V_{12} & V_{13} & W_x \\ V_{21} & V_{22} & V_{23} & W_y \\ V_{31} & V_{32} & V_{33} & W_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{V}_{3 \times 3} & \mathbf{W}_{3 \times 1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} = \mathbf{T}_{cm} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix}$$

**Figura 3.13:** Coordenadas de marca a coordenadas de cámara

Una vez se ha realizado el umbralizado de la imagen, se extraen las regiones que tienen un contorno que se puede delimitar con cuatro segmentos de línea. Los parámetros de estos cuatro segmentos de línea y las

coordenadas de los cuatro vértices de las regiones encontradas en las intersecciones de los segmentos de línea se guardan para su posterior proceso.

Las regiones se normalizan y la sub-imagen de la región se compara con los patrones dados anteriormente por el sistema y así poder identificar los identificadores de las marcas especificadas por el usuario. Para este proceso de normalización, se utiliza la ecuación 2 que representa la transformación perspectiva a utilizar. Todas las variables de la matriz de transformación se determinan substituyendo las coordenadas de la pantalla y las coordenadas de las marcas de los cuatro vértices de las marcas detectadas ( $x_c$ ,  $y_c$ ) y ( $x_m$ ,  $y_m$ ) respectivamente. Después se lleva a cabo el proceso de transformación utilizando esta matriz de transformación.

$$\begin{bmatrix} hx_c \\ hy_c \\ h \end{bmatrix} = \begin{bmatrix} N_{11} & N_{12} & N_{13} \\ N_{21} & N_{22} & N_{23} \\ N_{31} & N_{32} & 1 \end{bmatrix} \begin{bmatrix} X_m \\ Y_m \\ 1 \end{bmatrix}$$

Cuando dos lados paralelos de una marca cuadrada se proyectan sobre la imagen, las ecuaciones de estos segmentos de línea en las coordenadas de la cámara en la pantalla son las siguientes:

$$a_1x + b_1y + c_1z = 0, a_2x + b_2y + c_2z = 0$$

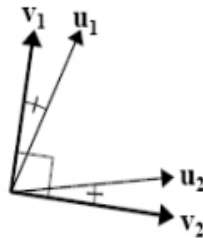
Para cada marca, el valor de estos parámetros se obtiene en el proceso de alineamiento adecuado. Dada la matriz de proyección perspectiva P que se obtiene con la calibración de la cámara en la ecuación 4, las ecuaciones de los planos que incluyen a estos dos lados respectivamente están representados en la ecuación 5 en el espacio de coordenadas de la cámara

$$P = \begin{bmatrix} P_{11} & P_{12} & P_{13} & 0 \\ 0 & P_{22} & P_{23} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \begin{bmatrix} hx_m \\ hy_m \\ h \\ 1 \end{bmatrix} = P \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}$$

substituyendo  $x_c$  e  $y_c$  en la ecuación 4 por  $x$  e  $y$  en la ecuación 3.

$$\begin{aligned} a_1 P_{11} X_e + (a_1 P_{12} + b_1 P_{22}) Y_e + (a_1 P_{13} + b_1 P_{23} + c_1) Z_e &= 0 \\ a_2 P_{11} X_e + (a_2 P_{12} + b_2 P_{22}) Y_e + (a_2 P_{13} + b_2 P_{23} + c_2) Z_e &= 0 \end{aligned}$$

Dados los vectores normales de estos planos que se llaman  $n_1$  y  $n_2$  respectivamente, el vector dirección de los dos lados paralelos del cuadrado se obtiene por el producto vectorial  $n_1 \times n_2$ . Así obtenemos dos vectores de dirección unitarios  $u_1$  y  $u_2$ , que pueden ser perpendiculares. No obstante, los errores en el procesamiento de la imagen significan que los vectores no serán exactamente perpendiculares. Para compensar estos dos vectores de dirección unitaria perpendiculares se definen como  $v_1$  y  $v_2$  en el plano que incluyen  $u_1$  y  $u_2$ . Dado que  $v_3$  es el vector de dirección unitaria que es perpendicular a  $v_1$  como a  $v_2$ , la componente de rotación  $V_{3 \times 3}$  en la matriz de transformación  $T_{cm}$  de las coordenadas de las marcas a las coordenadas de la cámara especificadas en la ecuación 1 es  $(v_1^t \ v_2^t \ v_3^t)$ .

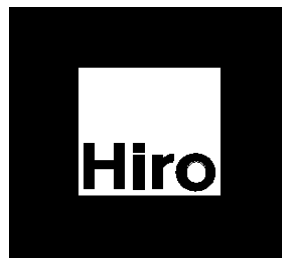


Una vez obtenidos estos valores, existen en realidad 8 ecuaciones de las cuales se obtiene el valor que falta para la componente de traslación  $W_{3 \times 1}$  ( $W_x$   $W_y$   $W_z$ ).

### 3.3.4.1 Marcador utilizado

La utilización de un patrón/marcador es otro de los aspectos imprescindibles en los sistemas de RA. En nuestro caso, no ha sido necesario definir un nuevo marcador ni generarlo, ya que se ha utilizado un marcador

genérico que ha sido empleado en otros sistemas, y es el que lleva ARToolKitPlus por defecto (figura 3.14), incluido en el fichero "patt.hiro".

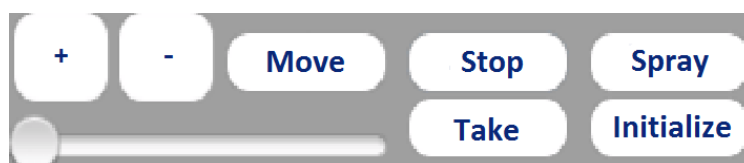


**Figura 3.14:** Marcador 2D utilizado

### 3.4 Funcionalidades

En esta sección veremos cuáles son las funcionalidades que se han implementado con respecto al proyecto original. Se ha propuesto para este sistema en iPhone que sea lo más similar posible, y ese es el objetivo que se ha intentado conseguir con las funciones que se han implementado. Debido a las limitaciones del iPhone, que aún siendo un dispositivo de gama alta, no se han podido realizar como en el original. Se ha seguido la ayuda de (LAMARCHE09).

Las funcionalidades implementadas han sido: iniciar escena, añadir/quitar objetos, movimiento, parada, simular un espray, modo de cucaracha estática-muerta y dos mejoras adicionales con respecto al proyecto base. Se ha implementado una interfaz gráfica con la herramienta Interface Builder, mediante botones táctiles, para dar soporte a estas funciones. La interfaz realizada puede ver en la figura 3.15. En cada descripción, se mostrará una captura del simulador de iPhone a modo de ejemplo donde se podrá ver en detalle cuál es el resultado de dicha acción, aunque en algunos casos el resultado, debido a que es una imagen estática, será el mismo.



**Figura 3.15:** Interfaz gráfica

### 3.4.1 Iniciar escena

Con esta acción se inicializa la escena, es decir, nos permite añadir objetos en la escena y nos sirve para en el momento que deseemos, borrar todos los objetos y volver al inicio, como si se tratase de una primera ejecución de la aplicación. También, nos sirve, para reiniciar la escena mientras otras funciones se están ejecutando, por ejemplo, en modo cucaracha muerta.

### 3.4.2 Añadir/Quitar objetos

El objetivo de estas funciones no es otro que el de permitir la inserción o eliminación de objetos de la escena, es decir, que se active la parte de dibujado de vértices de OpenGL ES. Como se ha podido leer en otros capítulos, tenemos dos tipos de cucarachas: cucaracha estática con posición uno y con posición dos. Representan el mismo tipo de objeto y lo único distinto entre ellos es la colocación de las patas. En el modelo uno, se han fijado de una manera, prestando atención a que esa colocación formase parte del posible movimiento que pudiese hacer una cucaracha real cuando está desplazándose. En el segundo modelo, cambia la posición de las patas, es decir, se han colocado de forma distinta al modelo uno. La explicación del porque de dos modelos estáticos se explicará en la parte del movimiento y animación, incluidos en este capítulo.

En cuanto a lo colocación de los objetos en pantalla, se ha establecido un valor aleatorio para cada objeto, tanto en la componente x como en la y. Se ha realizado una rotación sobre el eje z para que cada insecto aparezca con una orientación distinta, así cuando se inicia el movimiento cada insecto comienza a moverse en una determinada dirección, dando mayor realismo a la escena. En el proyecto original, el máximo de cucarachas que se podía insertar es de 60 insectos, que en nuestro caso, se ha establecido en un valor de 20 insectos, para no saturar el proceso de dibujado. Otra cuestión referida al proyecto base, es que existen tres grupos de cucarachas, cada uno con distintas posiciones y orientaciones según su colocación alrededor del

marcador. Se ha optado por incluir un único grupo de cucarachas, ya que las limitaciones de la pantalla de iPhone, nos impiden poder colocar más objetos alejados del marcador, porque si alejamos mucho la cámara del marcador, no se detecta el marcador y por lo tanto no se produce el dibujado de ningún objeto. Podemos ver la disposición que tiene los objetos en pantalla en la figura 3.16:



**Figura 3.16:** Ejemplo de colocación de objetos en la escena

### 3.4.3 Movimiento

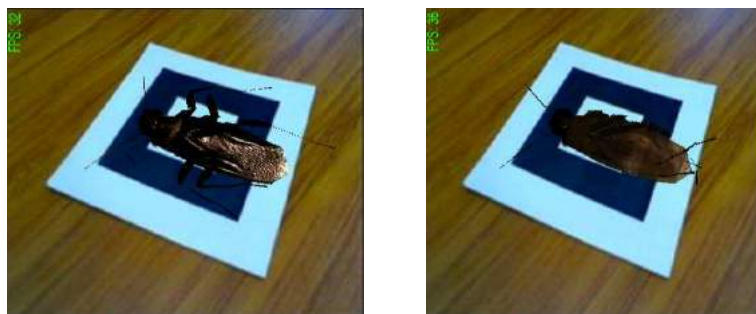
Éste es uno de los aspectos más importantes a la hora de dotar de realismo el conjunto de la aplicación. El movimiento de los objetos se ha realizado mediante el incremento de las posiciones en el eje y con respecto al patrón. En cada iteración de dibujado, se incrementa el valor del desplazamiento en y, por lo que traducido al entorno de programación de OpenGL ES, se realiza una traslación diferente en cada iteración. Cuando se ha llegado a un máximo fijado estáticamente en el programa, la cucaracha realiza un giro de 180º y comienza el movimiento desplazándose en sentido contrario al que se estaba desplazando.

El desplazamiento consiste en un avance de la cucaracha en línea recta, ida y vuelta, pasando siempre a una distancia cercana del patrón. Este desplazamiento se repite continuamente dentro de un bucle hasta que el usuario lo finalice mediante otra acción.

Junto con esta traslación, también se realiza un intercambio de objetos, tal como se comentó anteriormente. Se dispone de dos tipos de objetos cucaracha, con la diferencia de que las patas están ligeramente cambiadas en ambos modelos. Mediante este mecanismo, es posible que se vayan intercambiando los objetos para dar la sensación de que se está moviendo la cucaracha, cuando realmente se está produciendo un intercambio de objetos. El resultado final se aprecia de tal modo que realmente parece que es el mismo objeto, animado de modo conveniente.

### 3.4.4 Modo coger

Esta opción consiste en dejar solo una cucaracha muerta en el centro del patrón (ver figura 3.17). Se supone, que como en el mundo real, cuando una persona mata una cucaracha, lo normal, es luego deshacerse de ella y tirarla. En el caso, de nuestra aplicación sería exactamente lo mismo, si el usuario desea coger la cucaracha y tirarla, sólo tiene que coger este patrón y tirar la cucaracha. Para ello era necesario que no aparecieran el resto de cucarachas, tanto las vivas como las muertas, por lo que se optó como mejor opción dejar sólo una cucaracha muerta en el centro del patrón para poder deshacerse de ella.



**Figura 3.17:** Acción coger sobre la cucaracha



### 3.4.5 Simular espray

Otra funcionalidad que se ha implementado es la posibilidad de emular el sonido de un espray insecticida con el objetivo de dar realismo frente a una posible situación en la vida real. Esta acción, se puede realizar cuando los insectos se están moviendo por la pantalla, tal como podrían hacer los insectos si éstos fuesen auténticos y corriesen por el suelo. El efecto de esta acción es el de reproducir un sonido igual al que se produce al pulverizar con un insecticida. En la aplicación, una vez se ha reproducido el sonido, transcurren tres segundos hasta que las cucarachas empiezan a morir, es decir, se cambia de objeto 3D pasando de cucaracha estática (no importa si es el modelo 1 o el 2) a la cucaracha muerta. Está, mantiene todas sus translaciones en x e y, a fin de simular que el efecto del espray ha acabado con ella. Se ha optado por eliminar aquellos insectos cuyo orden de inserción en la escena es par, así conseguimos que si únicamente hay un objeto en la escena (impar), se produzca también el citado efecto. La figura 3.18 muestra como quedarían los objetos después de utilizar esta función:



**Figura 3.18:** Efecto de la opción Simular espray

### 3.4.6 Parada

Esta función tiene una lógica de funcionamiento muy simple. Consiste en suspender el incremento de la traslación, por tanto las cucarachas dejan de moverse, y se quedan en la posición en la que se encuentren, conservando las rotaciones (ver figura 3.19). Si al ejecutar esta función hubiese alguna cucaracha muerta en la escena, obviamente ésta seguirá quieta en su posición.



**Figura 3.19:** Cucarachas estáticas

### 3.4.7 Aumentar tamaño

Esta función nos sirve para agrandar las cucarachas que aparecen en la escena. Mediante la barra deslizante, aumentamos el tamaño de los objetos. Internamente se produce un incremento en una variable ligada a una función OpenGL. Se aplica a las cucarachas estáticas y a las que se mueven.

### 3.4.8 Movimiento mejorado

Se ha mejorado el movimiento de las cucarachas variando los incrementos y la dirección de traslación. En vez de realizar un movimiento recto en una única dirección, se ha modificado para que realice una pequeña rotación según avanzan en la dirección del eje Y, a fin de dotarlas de realismo, pues es como realmente se mueven este tipo de insectos. También se ha variado la velocidad de movimiento de un subconjunto de las cucarachas, concretamente las que se han insertado en la escena en orden impar poseen un movimiento más rápido que las insertadas en orden par.

### 3.4.9 Modo coger y eliminar

Esta opción consiste, al igual que el modo coger, en dejar sólo una cucaracha estática viva en el centro del patrón. En ese punto el usuario puede tocar la pantalla táctil para simular el hecho de eliminar el insecto. Si el usuario toca en la parte correspondiente de la pantalla, la cucaracha pasa del modelo estático al modelo de cucaracha muerta. Para dar más realismo al efecto, se ha añadido un sonido que simula un crujido, tal como ocurriría si pisamos un insecto de estas características. En este nuevo modo es necesario que no aparezcan el resto de cucarachas, tanto las vivas como las muertas, con tal de facilitar la acción al usuario, por lo que se ha optado por dejar sólo una cucaracha en el centro del patrón para poder deshacerse de ella.

# Capítulo 4

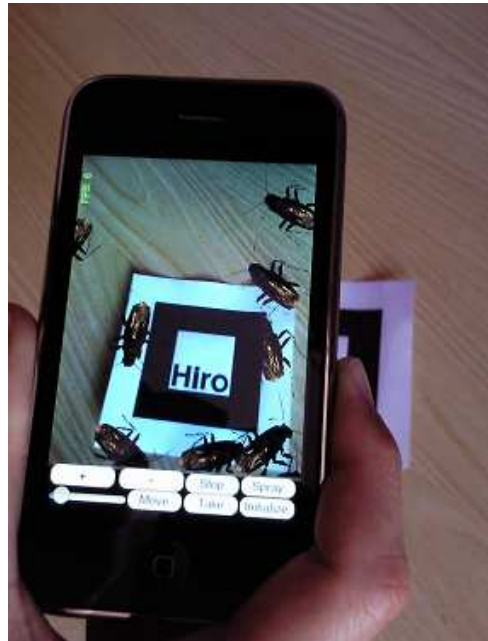
## Conclusiones y trabajos futuros

Inicialmente con el objetivo de conocer el estado del arte en dispositivos móviles, se elaboró un primer capítulo con dicho estado del arte. El objetivo final después de realizar ese estudio ha sido la implementación de una aplicación de RA para un dispositivo móvil, con el fin de ver qué posibilidades de desarrollo hay hoy en día para este tipo de desarrollos. Asimismo también ha servido para migrar un proyecto de RA que originalmente estaba diseñado para equipos de sobremesa, afianzando algunas de las hipótesis que se planteaban en el apartado 2.2.

En el segundo capítulo hemos podido observar en líneas generales cuál es el estado en que se encuentra la tecnología de RA en ese ámbito. Es sin duda con la aparición de nuevos dispositivos portátiles cuando la RA se vuelve móvil. Con el avance del hardware y del software de los dispositivos, los sistemas de RA diseñados para las plataformas tales como el PC, o los Tablet PC, serán válidos para ser implantados en los nuevos dispositivos.

Centrándonos en la aplicación desarrollada en el proyecto, se ha podido implementar cumpliendo los objetivos establecidos al comienzo del trabajo. Estos objetivos han sido la implementación y validación de la aplicación siguiendo unos criterios mínimos exigidos en cualquier trabajo de RA, como son la creación de una escena con objetos virtuales lo más realistas posibles, permitir una interacción entre el usuario y la escena creada y que todo ello sea en tiempo real, sin que se produzca un retraso visual perceptible por el usuario que está interactuando. Todas las funcionalidades desarrolladas realizan las mismas funciones que sus homólogas en el sistema de RA sobre PC, volviendo a demostrar que es posible el desarrollo de sistemas de RA sobre

dispositivos móviles (ver figura 3.20) y que éstos funcionen tan bien como en un equipo con mayores recursos.



**Figura 3.20:** Aplicación ejecutándose sobre iPhone

Tal como se comentó en otros capítulos, los avances en hardware posibilitan nuevas formas de interacción entre el usuario y la aplicación. Éste es el caso del dispositivo que se ha utilizado para desarrollar el proyecto, el iPhone. Cuenta con sensores capaces de dar información del entorno y de los movimientos que realiza el usuario cuando lo tiene en la mano. Una posible ampliación es el uso de los sensores como el acelerómetro para servir como apoyo en las funcionalidades, lo que aportaría una mayor implicación del usuario que está utilizando el programa. En una línea más general, podremos aprovechar el potencial de estos sensores para realizar sistemas de RA que aporten un mayor grado de interacción con el usuario que lo que aporta, por ejemplo, un PC. Esto es necesario en el campo del entretenimiento y el aprendizaje, pues se ha de captar la atención del usuario de manera que resulte atractivo para el manejo del sistema. Existen aplicaciones que ya han sido lanzadas al mercado general, haciendo uso de estos sensores, como se expuso en el apartado 2.7. Estamos pues, ante el comienzo del desarrollo de nuevos sistemas de RA móvil, que junto con la actual tecnología basada en marcadores, se desarrollarán de aquí en adelante.

# Capítulo 5

## Bibliografía

**(ARHH10)** ARhrr! Augmented Environments Lab. <http://www.augmentedenvironments.org/lab/research/handheld-ar/arhrrrr/> Última consulta: abril 2010

**(ARPLUS07)** ARToolKitPlus. Librería para la detección y seguimiento de marcadores 2D para dispositivos móviles.

[http://studierstube.icg.tu-graz.ac.at/handheld\\_ar/artoolkitplus.php](http://studierstube.icg.tu-graz.ac.at/handheld_ar/artoolkitplus.php). Última consulta: abril 2010

**(ARTOOL99)** ARToolKit. Librería para la detección y seguimiento de marcadores 2D. <http://www.hitl.washington.edu/artoolkit/> Última consulta: junio 2010

**(ART99)** H. Kato and M. Billinghurst, Marker tracking and HMD calibration for a video-based augmented reality conferencing system, Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR 99), 1999, pp. 85-94.

**(AZUMA97)** R. Azuma, "A survey of augmented reality", Presence: Teleoperators and Virtual Environments, 1997, pp. 355-385.

**(BILLINGHURST01)** Billinghurst, M., Kato, H., Poupyrev, I. (2001). The MagicBook: A Transitional AR Interface. Computers and Graphics, November 2001, pp. 745-753. Vol 25, N° 5

**(CANALYS09)** Situación actual de los SmartPhone. <http://www.canalys.com/pr/2009/200912.htm> Última consulta: diciembre 2009

**(CARRION04)** César Carrión Villaplana. Proyecto final de carrera: "Desarrollo de un sistema de Realidad Aumentada para el tratamiento de fobias a animales pequeños". Universidad Politécnica de Valencia. Director: M. Carmen Juan Lizandra. 2004

**(CAUDELL92)** T. P. Caudell, and D. W. Mizell, "Augmented Reality: An Application of Heads-Up Display Technology to Manual Manufacturing Processes", Proceedings of 1992 IEEE Hawaii International Conference on Systems Sciences, 1992, pp 659-669.

## Capítulo 5: Bibliografía

**(CONNECTED10)** Connected Education. Disponible en línea en: [http://www.connectededucation.com/index.php?option=com\\_content&view=article&id=190&Itemid=17](http://www.connectededucation.com/index.php?option=com_content&view=article&id=190&Itemid=17). Última consulta: junio 2010

**(FEINER93)** S. Feiner, B. MacIntyre, and D. Seligmann. "Knowledge-based augmented reality". *Communications of the ACM*, 36(7), July 1993, 52-62.

**(FRUEND01)** J. Freund, C. Geiger, M. Grafe and B. Kleinjohann, "The Augmented Reality Personal Digital Assistant", *Proceedings of the Second International Symposium on Mixed Reality (ISAR 2001)*, 2001.

**(HENDERSON07)** S. Henderson, S. Feiner. *Augmented Reality for Maintenance and Repair (ARMAR)*. Technical Report AFRL-RH-WP-TR-2007-0112, United States Air Force Research Lab, Jul 2007

**(HENRYSSON05)** A. Henrysson, M. Billinghurst, and M. Ollila, "Face to Face Collaborative AR on Mobile Phones", *Proceedings of the 4th IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR 05)*, 2005, pp. 80-89.

**(HMAR09)** D. Wagner. History of mobile augmented reality. Extraído de: <https://www.icg.tugraz.at/~daniel/HistoryOfMobileAR/>. Última consulta: mayo 2010

**(HOLLERER99)** T. Höllerer, S. Feiner, T. Terauchi, G. Rashid and D. Hallaway, *Exploring MARS: Developing indoor and outdoor user interfaces to a mobile augmented reality system*. *Computers and Graphics*, 1999, pp. 779-785. Vol 23.

**(IDEV10)** iPhone Development. Application development for the iPhone's SDK. 2010. Recuperado del sitio web de [iphonedevdevelopment.blogspot.com](http://iphonedevdevelopment.blogspot.com) <http://iphonedevdevelopment.blogspot.com/2009/05/opengl-es-from-ground-up-table-of.html>. Última consulta: abril 2010

**(INV10)** Videojuego Invizimals. Disponible en línea en: <http://www.invizimals.com>. Última consulta: junio 2010

**(ISWC97)** S. Feiner, B. MacIntyre, T. Höllerer, and T. Webster, "A touring machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment". *Proc. ISWC '97 (First IEEE Int. Symp. on Wearable Computers)*, October 13-14, 1997, Cambridge, MA. También en *Personal Technologies*, 1(4), 1997, pp. 208-217

**(JUAN05)** Juan, M.C., Alcañiz, M., Monserrat, Botella, C., Baños, R.M. and Guerrero, B. (2005) Using augmented reality to treat phobias, *IEEE Computer Graphics and Applications*. 25(6): 31-37

## Capítulo 5: Bibliografía

**(JULIER00)** S. Julier, Y. Baillet, M. Lanzagorta, D. Brown and L. Rosenblum, "BARS: Battlefield Augmented Reality System", NATO Information Systems Technology Panel Symposium on New Information Processing Techniques for Military Systems, 2000.

**(KALKUSCKH02)** M. Kalkusch, T. Lidy, M. Knapp, G. Reitmayr, H. Kaufmann and D. Schmalstieg, "Structured Visual Markers for Indoor Pathfinding", Proceedings of the First IEEE International Workshop on ARToolKit (ART02), 2002. pp. 8

**(LAMARCHE09)** J. LaMarche, D. Mark. Beginning iPhone Development: Exploring iPhone SDK. Apress. 2009.

**(LIESTOL09)** G. Liestol. Augmented Reality and Digital Genre Design: Situated simulations on the iPhone. Gunnar Liestol. Dept. of Media & Communication, University of Oslo. IEEE International Symposium on Mixed and Augmented Reality 2009. pp. 635-637

**(MARS97)** S. Feiner, B. MacIntyre, T. Höllerer and A. Webster, "A touring machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment", Proceedings of First IEEE International Symposium on Wearable Computers (ISWC '97), 1997, pp 74-81.

**(MILGRAM94)** P. Milgram and F. Kishino, "Taxonomy of Mixed Reality Visual Displays", IEICE Transactions on Information and Systems, 1994, pp. 1321-1329.

**(MIYASHITA08)** T. Miyashita, P. Meier, T. Tachikawa, S. Orlic, T. Eble, V. Scholz, A. Gapel, O. Gerl, S. Arnaudov, S. Lieberknecht. Symposium on Mixed and Augmented Reality. Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality. 2008. pp. 103-106

**(MOHRING04)** M. Möhring, C. Lessig and O. Bimber, "Video See-Through AR on Consumer Cell Phones", Proceedings of the 3th IEEE/ACM international Symposium on Mixed and Augmented Reality (ISMAR 04), 2004, pp. 252-253.

**(MULLONI07)** A collaborative and location-aware application based on augmented reality for mobile devices. Master Thesis 2007. Universidad de Udine. Alessandro Mulloni.

**(MULT08)** Multimedia intelligence. [http://www.multimediantelligence.com/index.php?page=shop.product\\_details&flypage=flypage.tpl&product\\_id=11&option=com\\_virtuemart&Itemid=84](http://www.multimediantelligence.com/index.php?page=shop.product_details&flypage=flypage.tpl&product_id=11&option=com_virtuemart&Itemid=84) Última consulta: enero 2010

**(REITMAYR01)** G. Reitmayr, and D. Schmalstieg, "Mobile Collaborative Augmented Reality", Proceedings of the International Symposium on Augmented Reality, 2001, pp. 114-123.



## Capítulo 5: Bibliografía

**(REYTMAR06)** G. Reitmayr and T. Drummond, "Going Out: Robust Model-based Tracking for Outdoor Augmented Reality", Proceedings of 5th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2006), 2006, pp. 109-118.

**(REKIMOTO95)** J. Rekimoto and K. Nagao, "The World through the Computer: Computer Augmented Interaction with Real World Environments", Proceedings of the 8th annual ACM symposium on User interface and software technology (UIST '95), 1995, pp. 29-36.

**(ROHS04)** M. Rohs and B. Gfeller, "Using Camera-Equipped Mobile Phone for Interacting with Real-World Objects", Advances in Pervasive Computing, 2004, pp. 265-271.

**(SCHMALSTIEG07)** D. Wagner, D. Schmalstieg, "ARToolKitPlus for Pose Tracking on Mobile Devices". Computer Vision Winter Workshop 2007, Austria, February 6-8, 2007. pp 125-134.

**(SCHMALSTIEG03)** D. Wagner and D. Schmalstieg, "First Steps Towards Handheld Augmented Reality", Proceedings of the 7th IEEE International Symposium on Wearable Computers (ISWC 03), 2003, pp. 127-135.

**(SUTHERLAND68)** I. Sutherland, "A Head-Mounted Three Dimensional Display", Proceedings of Fall Joint Computer Conference, 1968, pp. 757-764.

**(TIC08)** Informe Anual de la sociedad de la información en España 2008. Secretaría de estado para las telecomunicaciones y para la sociedad de la información. 2008. P. 114

**(TIMEWARP07)** I. Herbst, A. Kathrin, B. McCall, W. Broll, "TimeWarp: Interactive Time Travel with a Mobile Mixed Reality Game". International Conference on Human-Computer Interaction with Mobile Devices and Services. Proceedings of the 10th international conference on Human computer interaction with mobile devices and services. Amsterdam, The Netherlands. pp: 235-244

**(WAGNER07)** D. Wagner. Dissertation Thesis "Handheld Augmented Reality". Graz University of Technology. Institute for Computer Graphics and Vision. Director: D. Schmalstieg. 2007.

**(WIKITUDE09)** Wikitude World Browser. 2010. Disponible en línea en: <http://www.wikitude.org>  
Última consulta: mayo 2010

**(WOODS03)** E. Woods, P. Mason, M. Billinghurst. MagcMouse: an Inexpensive 6-Degree-of-Freedom Mouse. Proceedings of Graphite 2003. pp. 285-286