

UNIVERSIDAD POLITÉCNICA DE VALENCIA

Máster en Automatización e Informática Industrial



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Integración de realidad aumentada por visión en aplicaciones para un robot industrial

Proyecto final de máster
Presentada por: Julián Martínez Jiménez
Dirigida por: Martín Mellado Arteché
Valencia, Septiembre 2012

Agradecimientos

Este proyecto representa el fin de un año de esfuerzo y trabajo. Ha sido un periodo de tiempo complicado durante el cual mi familia y amigos han estado apoyando y animando cada una de las decisiones que he tomado.

Durante este tiempo he podido vivir muchas experiencias y en todas ellas, agradecer a mis amigos de antes, de ahora y de siempre, su apoyo incondicional. No es necesario que mencione a nadie, puesto que el nombre no es relevante. El hecho de estar a mi lado en cada momento, incluso estando lejos en algunas ocasiones, es lo que realmente hace que me sienta orgulloso de tenerlos cerca.

Quiero mencionar también a mi familia en este proyecto, ya que día a día se interesan por mis necesidades y preocupaciones. Mis hermanos son un referente para mí: por su trabajo, dedicación y forma de ser. Añadir a mis padres, que a su manera, hacen lo que está en su mano para ayudarme en todo.

Por último quiero dar las gracias a mi novia, que durante este periodo ha podido soportar mis momentos más desquiciantes y compartir mis ilusiones y alegrías.

Índice

1. Objetivos y Motivación	9
1.1. Objeto	9
1.2. Objetivo	9
1.3. Resumen	9
2. Introducción	10
2.1. Planteamiento del problema	10
2.2. Estructura del proyecto	11
3. Estado del Arte	12
3.1. Realidad aumentada	12
3.2. Elementos estructurales	13
3.3. Realidad aumentada móvil	14
3.4. Antecedentes	15
3.5. Dispositivos portátiles	17
3.6. Campos de aplicación	21
4. Entorno de trabajo	24
4.1. Robot de trabajo	24
4.2. Comunicación entre usuario y robot	25
4.2.1. Comunicación entre el dispositivo móvil y un computador convencional	25
4.2.2. Comunicación entre el computador convencional y la unidad de control	26
4.2.3. Comunicación entre la unidad de control y el robot SCARA	27
4.3. Entorno de trabajo y desarrollo	27
4.3.1. Dispositivo móvil táctil	27
4.3.2. Computador convencional	28
4.3.3. Robot industrial	28
4.4. Integración de la realidad aumentada sobre el proyecto	29
4.4.1. Biblioteca ARToolKit	29
4.4.2. Identificación de marcadores	30
4.4.3. Codificación de marcadores	31
4.4.4. Incorporación de marcadores al proyecto	32
4.4.5. Implementación en código de marcadores	33
4.4.6. Implementación del registro de los marcadores	33
5. Aplicación de colisiones	35
5.1. Detección de marcadores	35
5.1.1. Sistema de referencia	35
5.1.2. Implementación de visibilidad de los marcadores	36
5.2. Detección de colisiones	37
5.2.1. Implementación de detección de colisiones	37
5.2.2. Aproximación por esferas envolventes	38
5.3. Acción ante colisiones	39
5.4. Funcionamiento de la aplicación de colisiones	40

6. Aplicación de selección de objetos	42
6.1. Detección de marcadores	42
6.2. Detección de objetos	42
6.2.1. Implementación de detección de objetos	42
6.3. Movimiento al objeto	43
6.4. Funcionamiento de la aplicación de selección de objetos	43
7. Conclusiones	45
8. Trabajo Futuro	46
9. Bibliografía	47

Índice de figuras

1.	Continuo de Milgran	12
2.	Concepto de realidad aumentada	13
3.	Elementos para desarrollar una aplicación basada en realidad aumentada	13
4.	Evolución de las nuevas tecnologías	15
5.	Combinación de una escena real con objetos virtuales. Ronald Azuma [7]	15
6.	BARS (Battlefield Augmented Reality System)	16
7.	AR-Tennis	16
8.	Wikitude (Mobilizy)	17
9.	Reality Fighters (PS Vita de SONY)	17
10.	HMD (Head Mounted Display)	18
11.	PDA (Personal Digital Assistant)	18
12.	Smartphone	19
13.	Tableta (Tablet PC)	20
14.	Ultra Mobile PC	20
15.	PS Vita (SONY)	21
16.	Aplicaciones industriales con realidad aumentada	22
17.	Proyectos educativos con realidad aumentada	22
18.	Robot SCARA KR 5 R350	24
19.	Robot SCARA de KUKA, unidad de control y panel de control	25
20.	Comunicación entre usuario y robot.	25
21.	KRC Editor.	26
22.	Entorno de trabajo	27
23.	Tableta del entorno de trabajo	28
24.	Biblioteca ARToolKit en el proyecto	29
25.	Creación de marcadores	30
26.	Relación entre tamaño y distancia	30
27.	Marcas utilizadas en el proyecto	31
28.	Codificación de marcadores	32
29.	Inclusión de los ficheros .patt en el proyecto	32
30.	Implementación de la representación 3D de las marcas	33
31.	Visualización general de la realidad aumentada	34
32.	Sistema de referencia	35
33.	Matriz de transformación	36
34.	Separación entre cámara y marcas	37
35.	Cálculos vectoriales de los objetos del entorno	38
36.	Esferas envolventes	38
37.	Información sobre una colisión	39
38.	Acción ante una colisión	40
39.	Menú de la aplicación Kuka Control	41
40.	Detección de objetos	42
41.	Movimiento al objeto	43
42.	Selección de la aplicación en el menú de Kuka Control	44

1. Objetivos y Motivación

1.1. Objeto

Se trata de presentar un conjunto funcionalidades sobre un robot industrial empleando técnicas de realidad aumentada. Las aplicaciones desarrolladas se aplicarán sobre dispositivos móviles táctiles y se utilizarán para interactuar con el robot.

1.2. Objetivo

La finalidad del proyecto es poder controlar un robot industrial a través de un dispositivo móvil táctil, empleando la técnica de realidad aumentada como un elemento de sensorización. De esta forma se pretende poder controlar el robot, su posición y los obstáculos o elementos que están en su entorno de trabajo. También poder interactuar con el robot y los objetos del entorno desde la interfaz del dispositivo móvil.

1.3. Resumen

En la industria un robot es manejado desde sus paneles de control y con ellos interactuar con el fin de realizar las tareas deseadas. La idea es poder controlar un robot industrial a partir de cualquier dispositivo móvil táctil, independientemente si se trata de un teléfono móvil, una tableta u otros dispositivos.

Partiendo de ésta aplicación, la idea es integrar aplicaciones de realidad aumentada con la finalidad de poder controlar el robot en su área de trabajo y los elementos que puedan aparecer en él. Colocando las marcas necesarias e interactuando a través de la aplicación móvil, se pretende manejar el robot considerando los posibles obstáculos que puedan aparecer, detectando posibles colisiones y rectificando el movimiento en dicho caso. Por otra parte se ha realizado otra aplicación con la que desde la interfaz del dispositivo móvil, se puede identificar los diferentes elementos que aparecen en el área de trabajo del robot. Seleccionando el objeto deseado de entre los marcados, se puede mover de forma automática el robot a la posición del objeto y realizar la operación de manejo deseada sobre el mismo.

2. Introducción

El fundamento básico del proyecto es poder introducir nuevos elementos de sensorización en un entorno de trabajo robotizado. Habitualmente los robots industriales se manejan de forma automática a través de computadores o de forma manual mediante los paneles de control. Actualmente se emplean múltiples sensores para capturar la información del entorno, analizarla, interpretarla y responder ante las acciones necesarias. Entre los sensores más empleados, destacan: cámaras para el reconocimiento de formas, ultrasonidos, láseres...

En este proyecto se pretende dar un enfoque diferente, aplicando las nuevas tecnologías y empleando dispositivos móviles inalámbricos táctiles como una forma de interacción entre usuario y robot. Aplicando la realidad aumentada como elemento de sensorización y desarrollando aplicaciones reales, se pretende comparar la fiabilidad y seguridad de utilizar esta técnica en un entorno robotizado.

Hoy en día muchos usuarios disponen al alcance de la mano de dispositivos móviles muy potentes. Entre las tareas más destacadas reside el poder integrar la realidad aumentada como aplicación en cualquier dispositivo (siempre que cumpla los requisitos necesarios). De esta forma, se podrían utilizar las nuevas tecnologías para poder trabajar en un entorno robotizado, abaratando costes y haciendo la comunicación entre usuario y robot más sencilla e intuitiva.

Partiendo del proyecto inicial desarrollado por Ángel Fernández-Cañada [1], en el que mediante un dispositivo móvil táctil se puede establecer comunicación con un robot industrial y controlar los movimientos del mismo, se introducen los elementos de realidad aumentada con la finalidad de desarrollar dos aplicaciones integradas en el proyecto.

La primera aplicación consiste en un control de colisiones. En el proyecto inicial se puede controlar el robot y realizar movimientos, pero no se tiene en cuenta la detección de obstáculos u otros elementos del entorno. La idea es colocar las marcas necesarias en el entorno de trabajo para que, mientras desde la aplicación principal se gestionan los movimientos del robot, a su vez, se esté realizando un control de colisiones con los obstáculos del área de trabajo, avisando al usuario en caso de aproximación y rectificando el movimiento para evitar colisiones.

En la segunda aplicación se pretende realizar un análisis del área de trabajo del robot, detectando a partir del dispositivo móvil los objetos y elementos marcados que aparecen. Una vez detectados el usuario puede seleccionar el objeto marcado deseado y el robot se moverá hasta la posición del mismo. En esta aplicación el usuario puede utilizar el movimiento libre del robot y en cualquier instante seleccionar el objeto deseado. Además integra el mecanismo en el que cuando se pierde contacto visual entre la cámara del dispositivo y el objeto seleccionado, se cancela el movimiento del robot para proporcionar mayor seguridad y evitar movimientos indeseados.

2.1. Planteamiento del problema

En entornos industriales se utilizan múltiples captadores para obtener la información del entorno de trabajo, analizarla y procesarla. Se pretende utilizar la realidad aumentada, aplicada sobre dispositivos móviles táctiles, como un elemento de sensorización.

Con el avance de las nuevas tecnologías, se puede tener al alcance de la mano dispositivos móviles muy potentes y de pequeñas dimensiones. Con la potencia de los procesadores y utilizando los sensores que ya llevan integrados (cámaras, GPS...) se puede computar la información rápidamente y tomar las acciones adecuadas.

La realidad aumentada es un sistema que sobre el que se están desarrollando muchas aplicaciones actualmente debido a su simplicidad y potencia. Permite una gran interacción entre usuario y entorno. Por ello podría integrarse en un entorno de trabajo con el fin de reducir costes y hacer la comunicación entre usuario y robot más intuitiva.

Por un lado existen aplicaciones con diferentes funcionalidades que integran realidad aumentada. Por otra parte, también existen otro tipo de trabajos para el control de movimientos de un robot

desde dispositivos móviles. Partiendo de la unión de las dos ideas anteriores, se pretende desarrollar aplicaciones que permitan el control de un robot industrial desde un dispositivo móvil, integrando realidad aumentada.

El hecho de integrar realidad aumentada, permite controlar y detectar los objetos del área de trabajo del robot. Mediante la interacción, el usuario puede tomar las acciones oportunas para trabajar con los objetos u obstáculos con seguridad.

2.2. Estructura del proyecto

A continuación se muestra la estructura que presenta el proyecto, detallando de forma resumida el contenido del mismo.

- En la sección primera. *Objetivos y Motivación*. Se comentan cuales son las metas y la motivación para el desarrollo del proyecto.
- En la sección segunda. *Introducción*. Se explica de forma resumida en qué se basa el proyecto y el funcionamiento de las aplicaciones. También se expone el planteamiento del proyecto.
- En la sección tercera. *Estado del Arte*. Se realiza un estudio sobre la realidad aumentada. Se tratan campos como conceptos, dispositivos, campos de aplicación y sistemas desarrollados.
- En la sección cuarta. *Entorno de trabajo*. Es donde se realiza un análisis de las características y funcionamiento de los elementos de trabajo para desarrollar el proyecto. También la forma de establecer comunicación entre ellos.
- En la sección quinta. *Aplicación de colisiones*. Se expone el funcionamiento y desarrollo de la aplicación de colisiones. Se muestra la forma de integrar la realidad aumentada en la aplicación y su implementación.
- En la sección sexta. *Aplicación de selección de objetos*. Se expone el funcionamiento y desarrollo de la aplicación de selección de objetos. Se muestra la forma de integrar la realidad aumentada en la aplicación y su implementación.
- En la sección séptima. *Conclusiones*. Se comentan los resultados y las conclusiones obtenidas del proyecto.
- En la sección octava. *Trabajo Futuro*. Se explican los posibles campos de trabajo futuro para ampliar sobre el proyecto: nuevas ideas, añadir nuevas funcionalidades y formas de optimizarlo.

3. Estado del Arte

3.1. Realidad aumentada

La realidad aumentada es el término que se usa para definir una visión directa o indirecta de un entorno físico del mundo real, cuyos elementos se combinan con elementos virtuales para la creación de una realidad mixta en tiempo real. Consiste en un conjunto de dispositivos que añaden información virtual a la información física ya existente, es decir, añadir una parte sintética virtual a lo real.

No existe una definición única de realidad aumentada, aunque hay varias definiciones según algunas publicaciones. Paul Milgram [6] define la realidad aumentada sobre la base de un continuo llamado Continuo de Milgram.



Figura 1: Continuo de Milgram

Un entorno virtual se considera como algo completamente sintético donde los usuarios están completamente sumergidos, el entorno real se considera el lado contrario, integrado sólo por los objetos reales limitado por las leyes de la física.

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, se puede distinguir entre realidad aumentada y virtualidad aumentada, dependiendo de cual 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 realidad aumentada se encuentra más cerca del entorno real que para el entorno virtual en el Continuo de Milgram, por ello, puede ser vista como una versión extendida del entorno real, complementada por los objetos virtuales.

Ronald Azuma [7] menciona tres características que representan la realidad aumentada y aportan su definición. Por tanto, la realidad aumentada:

- Debe combinar elementos reales y virtuales.
- Debe ser interactivo en tiempo real.
- Debe estar registrada en 3D.

Considerando esta definición, se plantea uno de los problemas de la realidad aumentada. Los objetos virtuales tridimensionales y los 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 alineación (reconocimiento de patrones a través de marcadores), proporcionando así diferentes niveles de precisión.



Figura 2: Concepto de realidad aumentada

De una forma resumida, la realidad virtual se centra en proporcionar un entorno virtual al usuario, es decir, sumergir al usuario en un mundo totalmente virtual. Tanto la realidad aumentada como la realidad virtual, pretenden introducir al usuario en un entorno tridimensional. La diferencia reside en que mientras la realidad virtual sumerge al usuario en un mundo plenamente virtual, la realidad aumentada superpone los datos informáticos al mundo real sin sustituir la realidad física. Por tanto, la realidad aumentada es la incorporación de datos e información digital en un entorno real, por medio del reconocimiento de patrones que realiza mediante un software. Con la ayuda de la tecnología (aplicando visión por computador y reconocimiento de objetos) la información sobre el mundo real alrededor del usuario se convierte en interactiva y digital.

3.2. Elementos estructurales

Como requisitos para el desarrollo de una aplicación basada en realidad aumentada, es necesario disponer de unos determinados elementos y dispositivos.

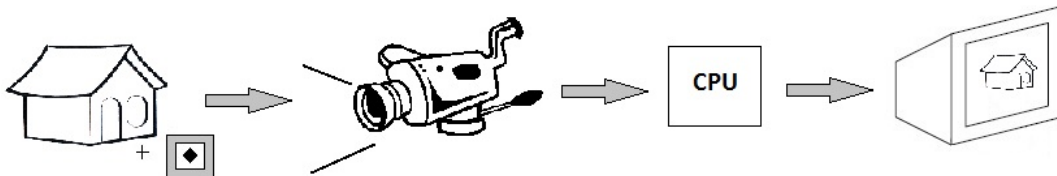


Figura 3: Elementos para desarrollar una aplicación basada en realidad aumentada

Tal y como se muestra en la figura anterior, son necesarios al menos cuatro elementos básicos:

- **Marcadores:** son plantillas, generalmente de papel o cualquier otro soporte plano, compuestas de símbolos. Se integran junto al mundo real y tras su interpretación por software, representarán una respuesta específica: mostrar imagen 3D, hacer cambios de movimiento al objeto creado...
- **Dispositivo de vídeo:** es indispensable para capturar la información obtenida del mundo real. Estos dispositivos pueden ser desde webcams integradas de bajo coste hasta cámaras de alta resolución.
- **Unidad de procesamiento:** se encarga de procesar e interpretar la información obtenida por el dispositivo de vídeo. Dependiendo de las necesidades de la aplicación, se pueden utilizar desde ordenadores de sobremesa, portátiles o tablets, hasta dispositivos móviles. Dependiendo de los requisitos hardware, se puede separar el trabajo de cálculo entre la CPU (registro de la realidad, lógica de la aplicación) y la GPU (renderizado de los objetos virtuales).
- **Dispositivo de visualización:** es el dispositivo sobre el que se mostrará la imagen resultado de la composición del mundo real y los objetos virtuales. Con las nuevas tecnologías aparecen nuevos

displays de visualización, aunque de forma más concreta suelen emplearse dos opciones básicas de representación.

- Tecnología óptica: esta tecnología se basa en la utilización de un combinador óptico situado en frente de los ojos del usuario. Este combinador es transparente, por lo que el usuario puede recibir información visual del mundo real y a su vez es capaz de reflejar el mundo virtual proyectado por un monitor. De este modo el dispositivo óptico únicamente representa los datos sintetizados (la imagen del mundo real se percibe por transparencia de las pantallas). El principal inconveniente de esta técnica es la reducción de luz que atraviesa la lente combinadora, por lo que las imágenes se oscurecen.
- Tecnología de vídeo: esta tecnología combina la imagen percibida del mundo real a través de una videocámara con los objetos sintetizados mediante un computador. El resultado se visualiza utilizando monitores convencionales. Estos monitores representan la imagen capturada del mundo real y la imagen virtual superpuesta.

3.3. Realidad aumentada móvil

Los primeros desarrollos hardware aplicados en realidad aumentada y muchos de los sistemas actuales se basan en equipos de sobremesa con cámaras fijas o en HMD (dispositivo de visualización parecido a un casco) con ordenadores portátiles incorporados. La principal ventaja de estos sistemas es que al ir adaptados a la cabeza permite disponer de las manos libres, aunque por contra, presentan los inconvenientes pertinentes como: altos costes, limitaciones de destreza, atractivo social...

Los sistemas de sobremesa fijos y los basados en mochila HMD presentan problemas de movilidad y por tanto, no son adecuados cuando se intenta integrar la realidad aumentada en aplicaciones portátiles. Un sistema móvil permite una mayor interacción con el usuario, ya que en cualquier instante y lugar, se puede hacer uso del mismo. En el proyecto se aplicará la realidad aumentada sobre una plataforma portátil y por ello se presta especial atención a este tipo de dispositivos.

Cuando se habla del concepto de movilidad, se hace referencia a dispositivos móviles que se pueden transportar con la mano (móviles, tablets, consolas portátiles, PDAs...). Hasta hace unos años, el hecho de poder realizar cómputo desde los dispositivos móviles era una tarea prácticamente imposible. La única forma de poder integrar la realidad aumentada en aplicaciones móviles era realizar todo el cálculo operacional de forma remota.

Con el avance de las nuevas tecnologías, han aparecido en el mercado procesadores integrados más potentes (CPU), procesadores de gráficos (GPU) y los últimos avances en capacidades inalámbricas y cámaras. Esto ha permitido considerar casi cualquier dispositivo móvil de hoy día como una plataforma de realidad aumentada independiente.

A pesar de de las capacidades de movilidad y las nuevas características hardware de estos dispositivos, también presentan algunas desventajas. Aunque la mejora tecnológica es considerable (evolución de gráficos 3D, conectividad, dimensiones de los dispositivos, aceleración de cómputo...), el consumo de energía es el principal limitante.

Por otra parte, 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. Por ello se ralentiza en un porcentaje muy elevado en comparación con la de punto flotante.

Otro factor a tener en cuenta son las dimensiones de pantalla de los dispositivos móviles y el pequeño campo de visión que limita el tipo de interacción con el usuario.



Figura 4: Evolución de las nuevas tecnologías

Considerando el paso de los años y la evolución de los sistemas de realidad aumentada, el desarrollo de aplicaciones sobre dispositivos móviles puede funcionar tan bien como los ordenadores de sobremesa. A pesar de sus desventajas, actualmente la mayoría de personas están familiarizadas con el uso de dispositivos portátiles. Dependiendo de las restricciones de las aplicaciones de realidad aumentada, puede ser un sistema estable y eficaz.

3.4. Antecedentes

En esta sección se expone de forma resumida las principales áreas de conocimiento e historia de la realidad aumentada, prestando especial atención a aquellos campos de investigación y aplicaciones sobre dispositivos móviles.

1968. Ivan Sutherland [9] crea el primer sistema de realidad aumentada. También es el inicio de la realidad virtual. Utiliza un sistema basado en HDM seguido por dos mecanismos para conseguir los seis grados de libertad. Debido al limitado poder de procesamiento de los ordenadores en aquella época, únicamente se podían crear dibujos sencillos alámbricos para mostrar en tiempo real.

1992. Caudell y Mizell [10] asocian el término de “realidad aumentada” para hacer referencia al hecho de añadir información generada por ordenador al mundo real. Proponen utilizar esta tecnología para aportar información adicional a los trabajadores de las fábricas.

1994. Milgram y Kishino [6] explican el concepto de realidad mixta, realidad aumentada y realidad virtual. Se comentan las etapas y transiciones mediante el Continuo de Milgram (1).

1997. Ronald Azuma [7] simboliza el primer estudio sobre realidad aumentada. En él, se explican los aspectos más importantes sobre la realidad aumentada y los identifica en: combinación de una escena real con objetos virtuales, interacción en tiempo real y registro tridimensional.

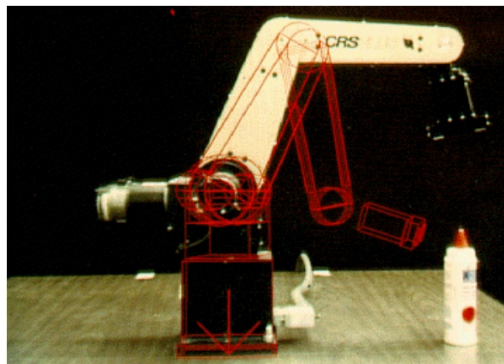


Figura 5: Combinación de una escena real con objetos virtuales. Ronald Azuma [7]

1999. Kato y Billinghurst [11] desarrollan ARToolKit. Se trata de una biblioteca de tracking de seguimiento de seis grados de libertad. Emplea marcadores bidimensionales cuadrados y reconocimiento. ARToolKit está disponible como código abierto bajo la licencia GPL y se utiliza hoy en día para el desarrollo de aplicaciones con realidad aumentada.

2000. Julier [12] se crea BARS (Battlefield Augmented Reality System). Se trata de un sistema HDM compuesto de una mochila con un ordenador y un conjunto de dispositivos. Entre otro está compuesto de sensores y receptores para visualizar información en un supuesto campo de batalla (por ejemplo: mapa de localización, información sobre los enemigos...).

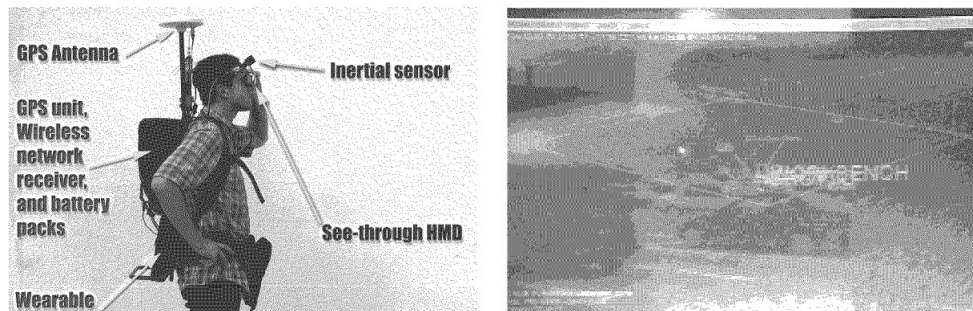


Figura 6: BARS (Battlefield Augmented Reality System)

2002. Kalkusch [13] desarrolla una aplicación de realidad aumentada sobre un HDM como guía para orientar a un usuario en el interior de un edificio. Se utiliza para orientar al usuario con la finalidad de alcanzar su destino. El sistema se basa en la superposición de un modelo alámbrico de la estructura del edificio. Emplean marcadores de ARToolKit.

2005. Henrysson [14] migra la librería ARToolKit para poder utilizarlo bajo el sistema operativo Symbian. Desarrollan AR-Tennis, la primera aplicación de realidad aumentada colaborativa para teléfonos móviles.

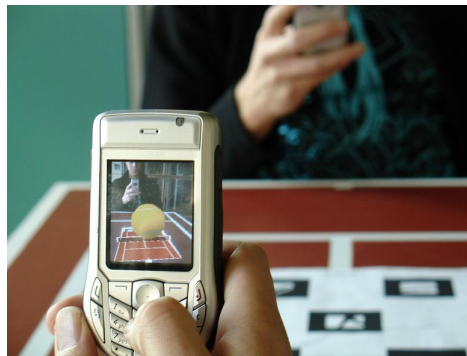


Figura 7: AR-Tennis

2006. Reitmayr [15] exponen un sistema de realidad aumentada aplicado para entornos urbanos, combinando otros dispositivos externos. En tiempo real permite capturar vídeo sobre un dispositivo PDA y visualizar el sistema.

2008. Mobilizy desarrolla Wikitude. Es una aplicación gratuita de realidad aumentada disponible para dispositivos Blackberry, iPhone, Windows y Android. Con las imágenes tomadas desde el dispositivo móvil, añade una capa de información: lugares, puntos de interés, restaurantes...

Combinando otros sensores tales como GPS, permite identificar si alguno de los contactos personales se encuentran alrededor de la ubicación actual, mostrándolo mediante realidad aumentada.

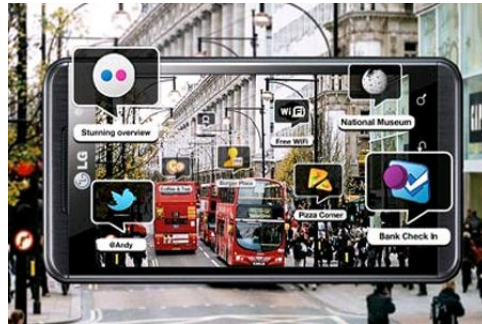


Figura 8: Wikitude (Mobilizy)

2012. Sony desarrolla Reality Fighters. Se trata de uno de los videojuegos más actuales basado en realidad aumentada, exclusivo para la consola portátil PS Vita.

Permite hacer una instantánea de la cara de cualquier persona o incluso de cualquier objeto y representa el rostro de un luchador. Integra otras funcionalidades adicionales como un micrófono para dar voz y personalidad al jugador.

Gracias a la realidad aumentada, permite luchar en cualquier escenario y contra cualquier jugador.



Figura 9: Reality Fighters (PS Vita de SONY)

Gracias a la evolución tecnológica y al estudio sobre la realidad aumentada, actualmente se pueden emplear nuevas técnicas y desarrollarlas en sistemas más potentes y portables. Esto permite una mayor interacción con el usuario, un mayor atractivo social y la aplicación de la realidad aumentada en nuevos campos de trabajo.

3.5. Dispositivos portátiles

Para trabajar con sistemas basados en realidad aumentada se pueden utilizar diferentes dispositivos. A lo largo de la historia, como se ha explicado en la sección 3.3, ha habido una evolución de los dispositivos, permitiendo integrar la realidad aumentada en la vida diaria. Antes, los sistemas que integraban esta técnica, debían ser sistemas fijos o con muy poca movilidad, limitando la funcionalidad y desarrollo.

Con el paso de los años, las nuevas tecnologías introducen nuevos elementos: sensores, cámaras, comunicación inalámbrica... Esto permite poder desarrollar aplicaciones con realidad aumentada y

utilizarlas en cualquier instante. Dependiendo de la finalidad y el campo de trabajo sobre el que se aplica, se pueden emplear los siguientes dispositivos:

HMD (Head Mounted Display)

Se trata de un dispositivo de visualización similar a un casco. Permite reproducir imágenes creadas por ordenador sobre un display muy cercano a los ojos o incluso directamente sobre la retina de los mismos (estos últimos se denominan monitor virtual de retina).

Un HMD puede ser de dos tipos. Si las imágenes creadas por computador se reproducen sobre un sólo ojo, se trata de un sistema monocular. De la misma forma, si la información se reproduce sobre los dos ojos, es binocular.



Figura 10: HMD (Head Mounted Display)

Las principal ventaja de este sistema es que engloba todo el campo de visión del usuario, sumergiéndolo en la escena mejor que con otros sistemas de pantallas. Al estar sujeto a la cabeza proporciona mayor libertad al usuario, integrándolo mejor en los ambientes creados por el ordenador.

Este sistema suele estar asociado a la conexión con un computador, por lo que proporciona una gran potencia de cálculo.

El principal inconveniente de estos sistemas es que son más caros y por ello no son los más extendidos.

PDA (Personal Digital Assistant)

Este dispositivo está diseñado como una agenda electrónica. Permite gestionar los contactos, fechas y calendarios, notas personales... Actualmente estos aparatos son más potentes y permiten realizar otras tareas como: reproducción de documentos y vídeos, gestión de correo electrónico, juegos...



Figura 11: PDA (Personal Digital Assistant)

La principal característica de estos sistemas es que son muy ligeros y tienen pequeñas dimensiones, por lo que son especialmente portables. Quizá tienen menor potencia de cómputo que otros sistemas pero son suficiente para trabajar con realidad aumentada.

Al ser sistemas más “simples”, en ocasiones requieren elementos externos para trabajar (cámaras, antenas para la conectividad...). Actualmente las PDAs están quedando relegadas a un segundo plano por la evolución de los Smartphones.

Smartphone

Un Smartphone (teléfono inteligente) es un teléfono móvil desarrollado sobre un sistema operativo móvil. Suponen la “evolución” de las PDAs introducida sobre un teléfono móvil convencional. Disponen de una gran capacidad de computación y conectividad. Disponen de múltiples aplicaciones que el usuario puede utilizar. Son uno de los sistemas más utilizados por su portabilidad y versatilidad.



Figura 12: Smartphone

Destacan sus dimensiones y su peso. Son ligeros y portables. Son muy potentes (muchos de ellos disponen de doble núcleo e incorporan funcionalidades gráficas especiales) y permiten procesar una gran cantidad de información como si de un ordenador personal se tratase. Integran tecnologías inalámbricas (Wi-Fi y Bluetooth), de localización (GPS), cámaras (algunas de ellas permiten tomar imágenes en 3D)... Incluye muchas aplicaciones, por lo que es muy atractivo para los usuarios.

El principal limitante de estos dispositivos es la autonomía. Al lanzar aplicaciones que requieren un cómputo elevado (o que requieren tecnologías inalámbricas y cámaras) la batería dispone de pocas horas de uso. Por otra parte, las dimensiones de la pantalla son reducidas para su utilización en campos de trabajo.

Tableta (Tablet PC)

Es un tipo de ordenador portátil integrado en una pantalla táctil con la que se interactúa principalmente con los dedos (según los modelos, se pueden también utilizar unos bolígrafos o plumas especiales). Son dispositivos de pequeñas dimensiones (manejaables con la mano sin ningún soporte adicional) y de poco peso.



Figura 13: Tableta (Tablet PC)

La principal ventaja de las tabletas, es que son sistemas ligeros y portables, permitiendo su utilización en cualquier instante. Permiten una gran interacción con el usuario. Hoy en día incluyen cada vez más elementos integrados (cámara, comunicaciones inalámbricas...) manteniendo una considerable duración de las baterías.

Dada la complejidad de la pantalla y los mecanismos que soportan la tecnología táctil hacen que la velocidad de interacción sea lenta (interpretación de movimientos y acciones). Considerando la igualdad de especificaciones hardware frente a un ordenador portátil, tienen un precio superior y una menor potencia de cómputo.-precio, la potencia de estos dispositivos también es menor.

Ultra Mobile PC

Se trata de un sistema híbrido entre una agenda electrónica personal, un ordenador portátil y una tableta. Disponen de una pantalla sensible al tacto y normalmente permite ejecutar con normalidad cualquier variante de Linux, Windows XP y Vista. Según las variantes, pueden disponer de teclado o no, así como otras funcionalidades adicionales.



Figura 14: Ultra Mobile PC

Destacan estos sistemas porque son lo más parecido a un ordenador portátil de pequeñas dimensiones. Por ello disponen de una gran capacidad de procesamiento y resolución de pantalla. Son de fácil transporte.

El problema es que la autonomía es muy limitada. Debido a sus características técnicas hace que el consumo de las baterías sea elevado.

Videoconsola portátil

Desde hace años la videoconsola ha sido disfrute en muchos hogares. Con el paso de los años empezaron a desarrollarse nuevas consolas portátiles. Inicialmente eran muy básicas, permitían jugar en dos dimensiones y en “blanco y negro”.

En la actualidad una videoconsola portátil es un dispositivo electrónico ligero que permite jugar a videojuegos integrando pantalla, altavoces y baterías en una misma unidad. Algunos modelos también incluyen cámaras y otros elementos de comunicación inalámbrica.



Figura 15: PS Vita (SONY)

Se han convertido en un dispositivo interesante de trabajo, ya que no tienen nada que envidiar a otros. Disponen de prestaciones similares o superiores que las PDA/Smartphone. Tienen un gran rendimiento y son muy útiles si se quieren utilizar en campos de trabajo sobre gráficos (incluyen FPU/GPU).

Pese a que se puede utilizar en ciertas áreas de trabajo concretas, su funcionalidad está limitada al entretenimiento. Algunos modelos poseen plataformas que no son estándar y dificultan su estudio y desarrollo.

3.6. Campos de aplicación

En esta sección se explica los diferentes campos de aplicación de la realidad aumentada. Cada vez más se está introduciendo esta técnica en la vida diaria, en el entorno de trabajo y otros servicios. A continuación se muestran los ejemplos más destacados.

Arquitectura

En el caso de la arquitectura puede ser muy práctico el poder visualizar edificios a gran escala y en tres dimensiones. De forma virtual se pueden alzar nuevas construcciones, edificios antiguos u otro tipo de diseños.

Aplicaciones industriales

Optar por la realidad aumentada en aplicaciones industriales es algo común hoy en día. Supone una muy buena elección cuando un trabajador ha de realizar alguna tarea de cierto riesgo frente a una máquina. Son un apoyo para la realización de tareas complejas.



Figura 16: Aplicaciones industriales con realidad aumentada

Por ejemplo, si se ha de manipular una máquina, con la realidad aumentada se pueden guiar los movimientos e indicaciones como medida de seguridad. Para llevar a cabo este tipo de procesos basta con emplear un casco que incorpore una cámara y un sistema adecuado de procesamiento de la información.

Educación

Muchas de las aplicaciones de realidad aumentada están destinadas hacia proyectos educativos. Los lugares más comunes son los museos, exposiciones u otro tipo de parques temáticos.

Uno de los ejemplos más interesantes es el poder mostrar un dinosaurio o cualquier otro animal a tamaño real a partir de una pequeña marca de realidad aumentada. Estas marcas pueden ser introducidas en libros de texto para mostrar la información más clara y mejorar el aprendizaje.



Figura 17: Proyectos educativos con realidad aumentada

Puede parecer algo simple, pero mostrar objetos aparentemente al alcance de la mano puede ser muy útil. Es una forma de interacción entre el usuario y el aprendizaje. Se puede conseguir envolver al usuario en ciertos escenarios: lugares, paisajes del pasado, ruinas antiguas...

Publicidad

Las empresas publicitarias buscan llamar la atención de los consumidores con ideas y formas más novedosas. Hoy en día, en muchos carteles publicitarios vienen reflejados marcas bidimensionales. Cuando éstas son interpretadas por cualquier dispositivo móvil, muestran información sobre un producto, una figura, un enlace a una página web o simplemente un texto.

Medicina

Otro campo importante de aplicación de la realidad aumentada es la medicina. Desde introducir marcas en cajas de fármacos para obtener información adicional hasta poder utilizar un simulador para cirugía. Permite tratar pacientes con una mayor seguridad y eficacia.

Informativos

Se trata de desarrollar software aplicando realidad aumentada por el simple hecho de aportar información adicional al usuario. En este sector se han creado algunas de las aplicaciones más importantes del mercado. Un ejemplo destacado es Wikitude, mencionado previamente en la sección 3.4, que permite dar información adicional sobre lugares (restaurantes, puntos de interés, contactos cercanos...).

Simulaciones

Es uno de los campos más destacados de aplicación de realidad aumentada. Para formar a un piloto, paracaidista o incluso para un entrenamiento militar específico se emplean técnicas de simulación. Aplicando la realidad aumentada se evitan arriesgar vidas durante el periodo de aprendizaje.

4. Entorno de trabajo

En esta sección se exponen los elementos de trabajo necesarios para desarrollar el proyecto. También se mencionan las especificaciones de cada uno de ellos y los aspectos técnicos.

4.1. Robot de trabajo

El robot con el que se trabaja es un robot SCARA KR 5 R350 proporcionado por la empresa KUKA. La empresa KUKA es uno de los principales fabricantes mundiales de robots industriales y sistemas de automatización para sectores que abarcan industrias tan diferentes como las del automóvil, metales, alimentos y plásticos. KUKA cuenta con más de 20 filiales en todo el mundo, siendo los líderes en el mercado mundial de robots industriales. Los robots industriales de KUKA se emplean sobre todo en fábricas para tareas de tratado de material, manipulación, carga y descarga de máquinas, paletizado, soldadura por puntos, embalaje, procesamiento y otros trabajos de automatización en diversas industrias.

En general, un robot SCARA (Selective Compliant Articulated Robot Arm) es un brazo articulado con tres ejes angulares (dos rotacionales y uno traslacional) y uno prismático. En total dispone de cuatro ejes que permite movimientos en los ejes X-Y-Z. Tiene un alcance de 350 mm con una capacidad de carga de 5 kg y una alta exactitud de posicionamiento de menos de 0,02 mm. El robot mismo necesita solamente una superficie de emplazamiento de pequeñas dimensiones (aprox. 150 x 150 mm) y pesa aproximadamente 20 kg.



Figura 18: Robot SCARA KR 5 R350

Son ideales siempre que se requiera gran precisión y velocidad. Consiguen efectuar con gran precisión las más complicadas tareas en espacios reducidos siendo al mismo tiempo los más rápidos de su clase. Por ello, representan la opción más acertada para tratar componentes pequeños de forma precisa y rápida.

Considerando los elementos físicos del robot, se compone de:

- Robot SCARA de KUKA: es el brazo robot que realizará las órdenes recibidas por la unidad de control o por el panel de control.
- Unidad de control: funciona como un servidor, se trata de una cpu que recibe las órdenes de usuario, las transforma en órdenes del robot y las envía al robot.
- Panel de control: va conectado a la unidad de control. Permite introducir la línea de comandos para las tareas del robot así como cargar y guardar programas. En el proyecto no se empleará este elemento, ya que el envío de órdenes se realizará desde el dispositivo móvil inalámbrico táctil.

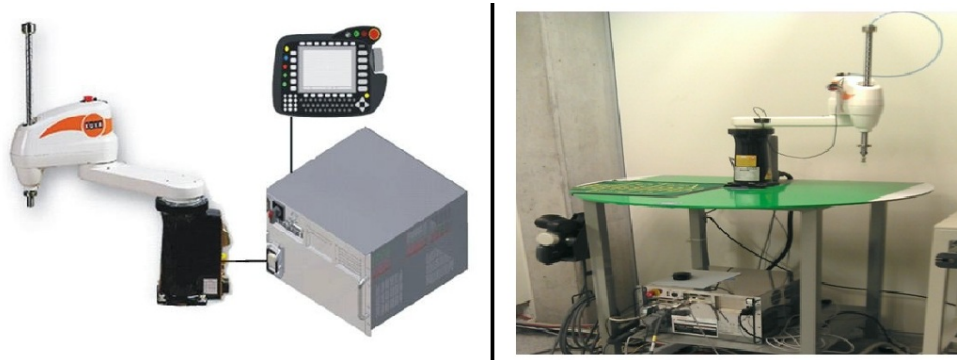


Figura 19: Robot SCARA de KUKA, unidad de control y panel de control

4.2. Comunicación entre usuario y robot

Lo usual en una comunicación entre usuario y robot es realizarla en 3 pasos. En primer lugar el usuario, a través del panel de control, introduce las órdenes o los programas deseados. Seguidamente, de forma transparente, las órdenes se reciben por la unidad de control, se transforman en órdenes del robot y se envían al robot. Por último, una vez recibidas las órdenes correspondientes al robot, se ejecutan los movimientos y operaciones deseadas.

En el proyecto se ha considerado otra forma de trabajar ya que se pretende poder controlar el robot a partir de un dispositivo móvil. Continuando con el proyecto de Ángel Fernández-Cañada [1] la comunicación entre usuario y robot se establece en las siguientes etapas, tal y como se muestra en la siguiente figura.

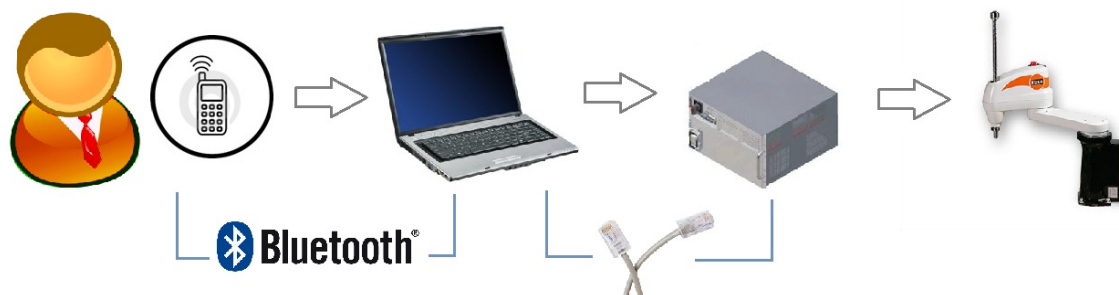


Figura 20: Comunicación entre usuario y robot.

4.2.1. Comunicación entre el dispositivo móvil y un computador convencional

Inicialmente el usuario, a partir del menú de la aplicación, selecciona el computador con el que se va a comunicar el dispositivo móvil. La idea es que el computador convencional funciona como un servidor. Vía la comunicación inalámbrica Bluetooth, se encarga de recibir cada una de las órdenes que el usuario introduce.

El dispositivo móvil táctil funciona como si de un panel de control se tratase. Mediante movimientos táctiles y la configuración de la aplicación, el usuario introduce los movimientos que el robot deseados. La aplicación interpreta los movimientos realizados y seguidamente envía las órdenes por Bluetooth al computador convencional. De forma transparente, el usuario envía al computador las órdenes con el siguiente formato.

```

"<MOD0>" modo "</MOD0> \n "
"<ExternalData> \n <Position> \n\t <A1>" valorEje1
"</A1> \n\t <A2>" valorEje2
"</A2> \n\t <A3>" valorEje3
"</A3> \n\t <A4>" valorEje4
"</A4> \n </Position> \n <Ventosa> \n\t <Activada>" herramientaAct
"</Activada> \n </Ventosa> \n <Speed> \n\t <Valor>" velocidad
"</Valor> \n </Speed> \n </ExternalData> \n"

```

Como se puede observar, con este formato se puede representar toda la información de los movimientos del robot, teniendo en cuenta el modo de trabajo, los ejes que se desea mover, la velocidad de movimiento e incluso si se desea activar la herramienta del robot. Toda esta información se envía al computador como una orden en cada movimiento táctil introducido por el usuario.

4.2.2. Comunicación entre el computador convencional y la unidad de control

En este punto, el computador convencional ha recibido las órdenes del usuario. Considerando el proyecto de Pablo Muñoz [2] en el que se desarrollan las aplicaciones de interpretación y envío de órdenes, se trabaja con una aplicación que “desgrana” los campos de información de la orden recibida y se interpreta cada uno de los campos.

Para continuar con el envío de órdenes, es necesario establecer comunicación entre el computador y la unidad de control. La unidad de control funciona principalmente como una unidad servidora, se dedica a escuchar las órdenes que le llegan a partir de una tarjeta de red integrada. A partir de otra aplicación del computador convencional, se definen los parámetros para la comunicación: dirección IP y puerto. Se emplea un cable Ethernet para transferir las órdenes entre el computador y la unidad de control.

Dado que la unidad de control ha de interpretar las órdenes que se envían, se emplea un programa de desarrollo llamado KRC Editor, en él se identifican los parámetros y valores aplicados al robot. Se indica qué valores se asignan a cada eje del robot, así como otros parámetros: velocidad, herramienta...

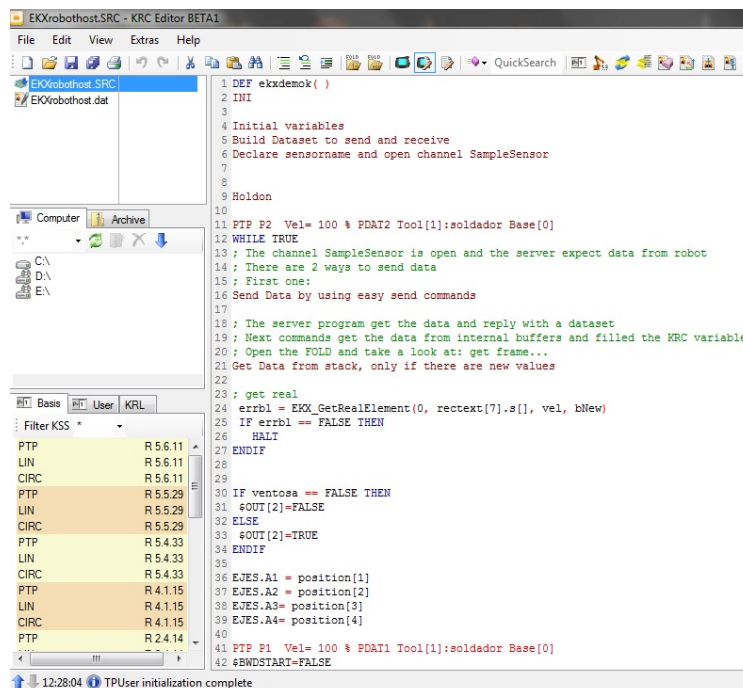


Figura 21: KRC Editor.

Tras emplear aplicar el código por el KRC Editor, la unidad de control ya puede interpretar los movimientos y las órdenes deseadas.

4.2.3. Comunicación entre la unidad de control y el robot SCARA

Una vez la unidad de control ha recibido las órdenes como espera interpretar, tan sólo tiene que dar las órdenes adecuadas al robot. Sobre la unidad de control no se ha modificado nada, toda la interpretación de las órdenes del usuario, la codificación y el envío, se realiza entre las dos etapas anteriores.

4.3. Entorno de trabajo y desarrollo

El entorno de trabajo está formado por tres elementos: el dispositivo móvil táctil, el computador convencional y el robot (robot SCARA y unidad de control).

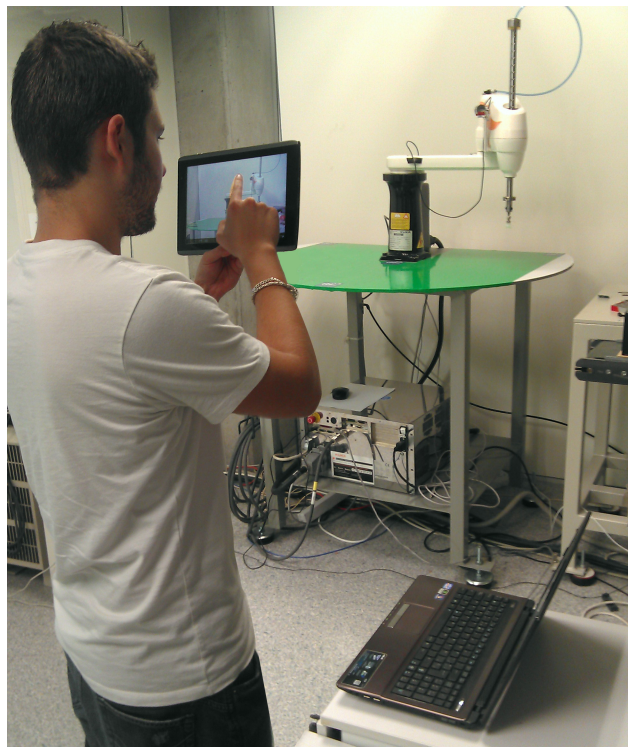


Figura 22: Entorno de trabajo

4.3.1. Dispositivo móvil táctil

Respecto al dispositivo móvil táctil, se trabaja con una tableta ACER ICONIA TAB A500. Entre sus características más destacables constan: su velocidad de procesador de 1GHz, memoria RAM DDR2 de 1GB, pantalla de 10 pulgadas con una resolución de 1280x800, cámara de alta definición de 1080p, Bluetooth y una versión de Android 3.0.



Figura 23: Tableta del entorno de trabajo

Durante la realización del proyecto se ha ido desarrollando y comprobando el funcionamiento con otros tipos de dispositivos móviles, no es necesario que se trate de una tableta, basta con cualquier dispositivo móvil táctil con cámara, sistema operativo Android, comunicación inalámbrica Bluetooth, una pantalla adecuada para poder realizar los movimientos táctiles y una suficiente velocidad de procesamiento para llevar la ejecución de la aplicación.

La idea de la tableta es situarla con un trípode o soporte justo en frente del robot, con una separación adecuada para abarcar el entorno. Dada su amplia pantalla, el usuario puede ver todo área de trabajo del robot desde el dispositivo a través de la aplicación. Así el usuario puede realizar los movimientos táctiles oportunos para controlar el robot teniendo en cuenta el entorno de trabajo del robot.

Desde el punto de vista del desarrollo la aplicación está implementada como un proyecto en Java. Se utiliza el entorno de desarrollo de Eclipse ya que permite la opción de incluir las librerías necesarias de Android, para trabajar con cierta facilidad.

4.3.2. Computador convencional

Por parte del computador convencional, simplemente es un elemento de comunicación. Como se ha explicado en el punto anterior, tan sólo se encarga de recibir las órdenes del dispositivo móvil, “desgranar” la información y traducirla en un formato adecuado para su posterior envío a la unidad de control. Se puede emplear cualquier equipo siempre que soporte la plataforma Windows (XP o superior).

Al igual que se menciona en el punto anterior, el elemento de desarrollo del computador convencional principalmente es el KRC Editor, para indicar al robot los valores de movimiento. Las otras aplicaciones más sencillas de traducción de información y comunicación TCP/IP, están desarrolladas en Java bajo el entorno de Eclipse.

4.3.3. Robot industrial

El último elemento del entorno de trabajo lo forman el robot y la unidad de control. Podría decirse que es el elemento más importante ya que es el que va a ejecutar las acciones del usuario, sin embargo es el único elemento sobre el que no se modifica nada. Como se ha explicado en el punto anterior, todas las traducciones de órdenes y envíos se realizan en los elementos previos de la comunicación. La unidad de control tan sólo traduce las órdenes del formato adecuado en valores de voltaje sobre el robot SCARA y en algunos casos puede proporcionar información del estado del robot.

4.4. Integración de la realidad aumentada sobre el proyecto

En este proyecto se ha introducido la realidad aumentada como un elemento de sensorización sobre el entorno de trabajo del robot. Como ampliación sobre el proyecto de Ángel Fernández-Cañada [1], se pretende aplicar esta técnica aportando un nuevo enfoque.

En el proyecto original se puede controlar un robot industrial desde un dispositivo móvil táctil sin ninguna restricción, es decir, el usuario puede manejar el robot sin considerar los elementos u obstáculos que puedan haber en el área de trabajo. Integrando la realidad aumentada y colocando las marcas sobre los elementos que interesen, se puede “sensorizar” el entorno. Considerando ésto último, se añaden dos funcionalidades: el control de colisiones y la captura de objetos marcados.

A continuación se muestran los pasos para la integración de la realidad aumentada sobre el proyecto.

4.4.1. Biblioteca ARToolKit

Primero es imprescindible introducir sobre el proyecto las bibliotecas necesarias para trabajar con realidad aumentada. Actualmente existen múltiples librerías dependiendo el lenguaje de programación sobre el que se va a trabajar y sus aplicaciones, entre ellas destacan: SlartoolKit (librería en visual C#), Ostrich (librería AS3 para detección de movimiento), HandyAR (librería para crear realidad aumentada con detección de manos)...

En el caso del proyecto se utiliza la biblioteca ARToolKit. Esta biblioteca permite la creación de aplicaciones de realidad aumentada, en las que se sobrepone imágenes virtuales al mundo real. Para ello, utiliza las capacidades de seguimiento de vídeo, con el fin de calcular en tiempo real, la posición de la cámara y la orientación relativa a la posición de los marcadores físicos. Una vez que la posición de la cámara real se sabe, los modelos 3D son sobrepuestos exactamente sobre el marcador real. Así ARToolKit resuelve dos de los principales problemas en la realidad aumentada, el seguimiento de punto de vista y la interacción objeto virtual.

En concreto se emplea las funcionalidades del proyecto AndAR integradas sobre la biblioteca ARToolKit. AndAR (Android Augmented Reality) es un proyecto que integra la realidad aumentada sobre la plataforma Android. Actualmente se mantiene como código abierto y se emplea en muchas ocasiones para aplicaciones móviles con sistema operativo Android, como es en nuestro caso. Basta con descargar la versión más adecuada e integrarla en el proyecto Java para Android.

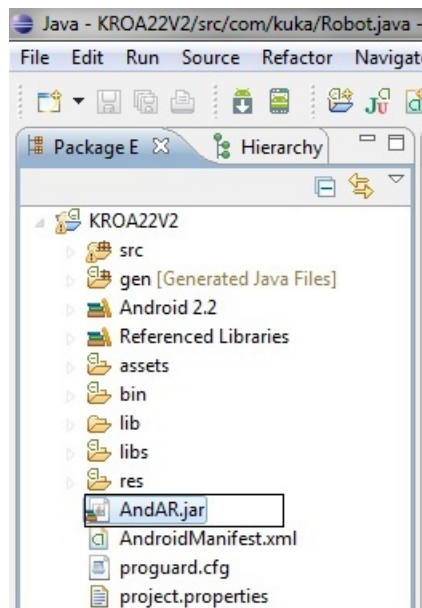


Figura 24: Biblioteca ARToolKit en el proyecto

Dado que se trata de código abierto, muchos usuarios incluyen nuevas versiones, ampliaciones o modificaciones sobre el código original, haciéndolo más eficiente y funcional para la aplicación en concreto.

4.4.2. Identificación de marcadores

Para poder trabajar con realidad aumentada es necesario crear, identificar y utilizar marcadores. Un marcador o patrón es una imagen (generalmente impresa en soporte de papel o similar) que el computador (dispositivo móvil) procesa y de acuerdo a la programación definida para esa imagen, incorpora objetos 3D. Los marcadores para realidad aumentada están realizados generalmente en archivos con formato de imagen y en un archivo .pat o .patt que guarda la codificación de dicha imagen.



Figura 25: Creación de marcadores

Para crear un marcador basta con identificar un símbolo e insertarlo en el interior de un recuadro de las medidas adecuadas, tal y como se observa en la figura anterior. Entre los factores para una visualización adecuada de la marca, destacan la luz y la distancia. Es indispensable tener suficiente luz en el entorno para que se pueda realizar la identificación y además se ha de tener las proporciones adecuadas entre la separación de la cámara y el tamaño del marcador. A continuación se muestra la relación descrita.

Tamaño	Distancia
16 x 16 cm	1,90 m
11 x 11 cm	1,30 m
7,5 x 7,5 cm	1,00 m
3,5 x 3,5 cm	0,40 m
1,3 x 1,3 cm	0,16 m

Figura 26: Relación entre tamaño y distancia

La relación mostrada anteriormente es orientativa. Dependiendo de la calidad de la cámara y la simplicidad de las marcas creadas se puede ajustar más o menos el tamaño de la marca, acorde a la distancia de separación.

Para el desarrollo del proyecto se han considerado siete marcas: tres relacionadas con los objetos del entorno y cuatro para representar la información del robot.

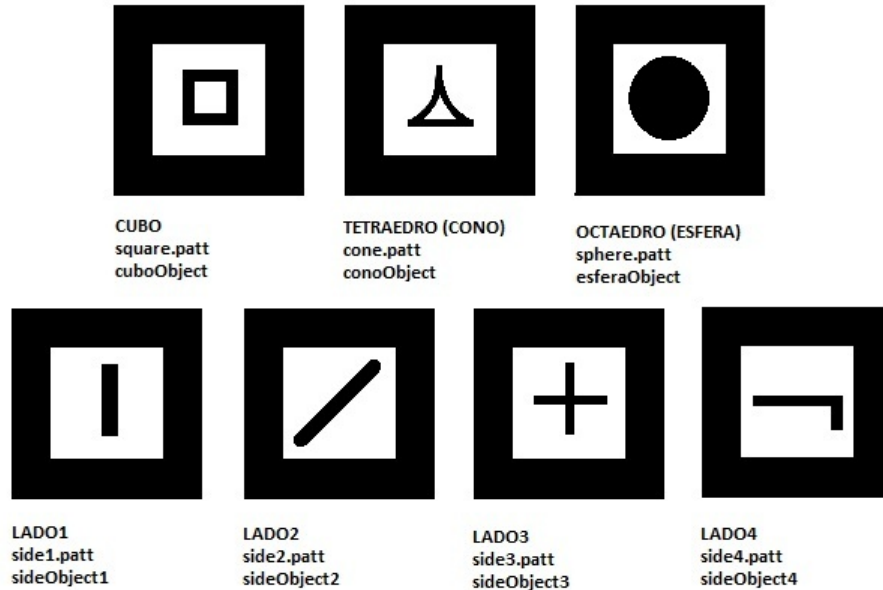


Figura 27: Marcas utilizadas en el proyecto

Teniendo en cuenta la figura anterior, las tres primeras marcas son para representar los objetos del entorno. Representan respectivamente las figuras tridimensionales de: un cubo, un tetraedro (cono) y un octaedro (esfera).

Las otras cuatro marcas son para colocarlas sobre la herramienta o el eje del robot, así se puede determinar la posición del robot. En concreto se emplean cuatro marcas ya que si el robot gira en alguno de sus ejes y se pierde la visualización de una marca, con cualquiera de las tres restantes, el robot sigue identificado y posicionado. La representación tridimensional de las cuatro marcas es la misma, un rombo.

Se ha tenido en cuenta las condiciones de luminosidad del laboratorio, la calidad de la cámara de la tableta, así como la separación entre la cámara y las marcas. Por ello, las marcas que representan los objetos del entorno tienen unas dimensiones de 8.5 x 8.5, mientras que las del robot son de 5 x 5 (más pequeñas para poder colocarlas sobre la herramienta o el eje del robot). La separación entre la cámara y las marcas es de 1.5 metros aproximadamente.

4.4.3. Codificación de marcadores

Una vez definidas las marcas que se van a utilizar, se han de identificar y obtener la codificación de las imágenes. Para ello se utiliza un conjunto de aplicaciones proporcionadas entre las descargas de AndAR [21]. Entre ellas, “mk_patt.exe” permite la creación de los ficheros .patt o .pat con la información de las plantillas codificadas.

La forma de hacer la codificación es muy sencilla. Primero se imprime en papel la plantilla que se quiere codificar. Seguidamente se abre la aplicación donde se activa la webcam del computador. Se le muestra a la webcam la marca, y con un sólo click, se obtienen los ficheros .patt de la marca codificada.



Figura 28: Codificación de marcadores

La codificación de las marcas componen ficheros con valores enteros entre 0 y 255. Al igual que se emplea esta aplicación se pueden emplear otras, ya que el objetivo principal es simplemente obtener el fichero .patt con la codificación.

4.4.4. Incorporación de marcadores al proyecto

Una vez obtenida la codificación de las marcas, falta incorporar los ficheros. Para ello tan sólo hay que incluirlos en el directorio apropiado del proyecto.

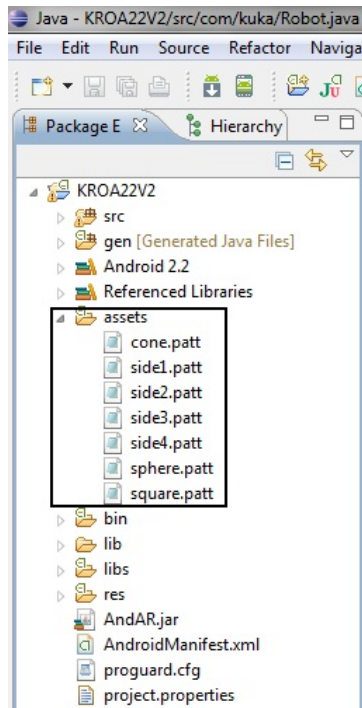


Figura 29: Inclusion de los ficheros .patt en el proyecto

Como se observa en la figura anterior, todos los ficheros .patt respectivos a la codificación de las marcas se han incluido en la carpeta “assets”. Una vez hecho esto, es cuando desde el código implementado se realiza la interpretación y asignación de las marcas.

4.4.5. Implementación en código de marcadores

Ahora es necesario implementar la representación tridimensional de cada una de las marcas. Para ello se crea una clase para cada una de las marcas.

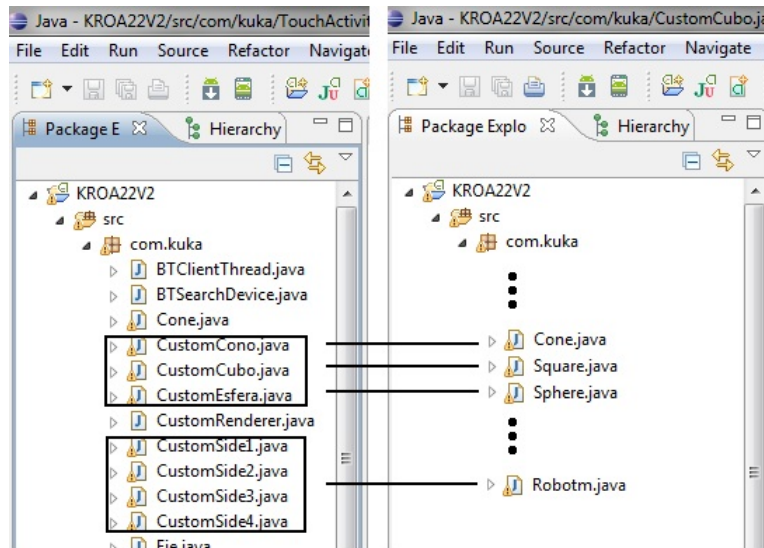


Figura 30: Implementación de la representación 3D de las marcas

En las clases “Cone.java”, “Square.java”, “Sphere.java” y “Robotm.java” es donde se implementa la forma geométrica de los objetos 3D. En estas clases se indican los índices de cada uno de los vértices de la figura a formar, la conexión entre ellos y el tamaño (alto, ancho y largo).

Seguidamente, en las clases “CustomCono.java”, “CustomCubo.java”, “CustomEsfera.java” y “CustomSide.java” es donde se realiza la invocación de la visualización de las clases anteriores. En estas clases además de la invocación, se indica también ciertos parámetros de visualización, tales como: color, efectos de luz (especular, difusa, ambiental), transparencia ...

En general las versiones de AndAR habituales permiten generar cualquier imagen 3D sobre las marcas. En nuestro caso particular, se ha elegido una versión que no permite esta opción, si no que permite generar figuras tridimensionales “más sencillas” formadas a partir de triángulos. El objetivo de esta elección es poder trabajar en tiempo real con las marcas a través del dispositivo móvil con el mínimo coste computacional y con la mayor fluidez posible.

Dado que en el entorno de trabajo del robot se van a utilizar múltiples marcas a la vez, en el proyecto interesa que el cómputo se utilice en los cálculos matemáticos de las aplicaciones realizadas y no en la generación de objetos tridimensionales complejos. De ahí, que la representación de un cono y de una esfera se realice como un tetraedro y un octaedro respectivamente.

4.4.6. Implementación del registro de los marcadores

Hasta este punto, se han definido las marcas, obtenida su codificación, se ha definido la forma geométrica de las figuras 3D y se han configurado los parámetros de visualización. Falta indicar desde el código implementado de la aplicación principal, la relación entre ellos.

```

artoolkit = super.getArtoolkit();

cuboObject = new CustomCubo("square", "square.patt", 85.0, new double[]{0,0});
artoolkit.registerARObject(cuboObject);

conoObject = new CustomCono ("cone", "cone.patt", 85.0, new double[]{0,0});
artoolkit.registerARObject(conoObject);

esferaObject = new CustomEsfera ("sphere", "sphere.patt", 85.0, new double[]{0,0});
artoolkit.registerARObject(esferaObject);

sideObject1 = new CustomSide1("side1", "side1.patt", 50.0, new double[]{0,0});
artoolkit.registerARObject(sideObject1);

sideObject2 = new CustomSide2("side2", "side2.patt", 50.0, new double[]{0,0});
artoolkit.registerARObject(sideObject2);

sideObject3 = new CustomSide3("side3", "side3.patt", 50.0, new double[]{0,0});
artoolkit.registerARObject(sideObject3);

sideObject4 = new CustomSide4("side4", "side4.patt", 50.0, new double[]{0,0});
artoolkit.registerARObject(sideObject4);

```

Esta es la tarea más sencilla, tan sólo se registran los objetos en el sistema con los parámetros adecuados. Primero se instancia un nuevo objeto de la clase que habíamos creado previamente, se le indica un nombre, el fichero de la marca codificada para su interpretación y el tamaño real de la marca en milímetros. Por último, se le indica al sistema el registro de los objetos deseados.

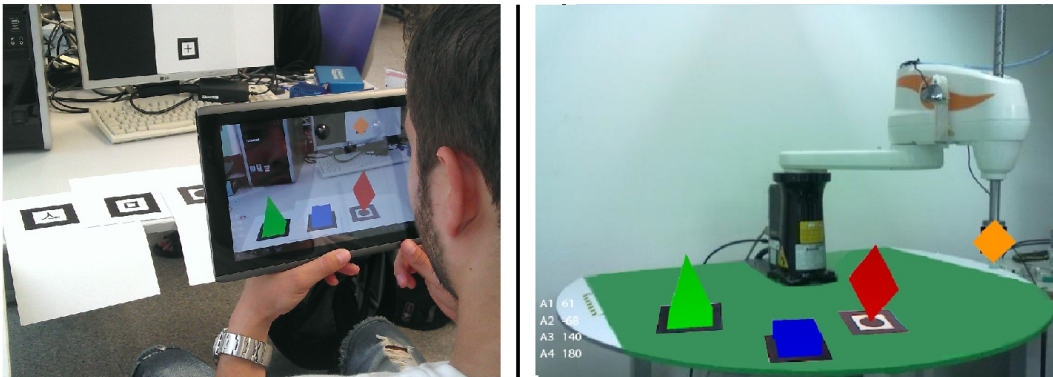


Figura 31: Visualización general de la realidad aumentada

5. Aplicación de colisiones

En esta sección se explica el funcionamiento y desarrollo de la aplicación de control de colisiones, integrada sobre el proyecto de Ángel Fernández-Cañada [1]. De forma resumida, se integra la realidad aumentada con la finalidad de poder controlar el robot industrial con total libertad.

Al integrar marcadores sobre los elementos de interés del entorno, el usuario puede mover el robot teniendo en cuenta los posibles obstáculos que puedan aparecer. De forma automática por el funcionamiento de la aplicación, ante un riesgo de colisión, se notifica al usuario y se corrige la trayectoria.

5.1. Detección de marcadores

Como se ha explicado en la sección 4.4.2, se utilizan siete marcadores. Cuatro de ellos son fijos y forman parte de la identificación y posicionamiento relativo del robot. Los otros tres marcadores son variables y hacen referencia a objetos u obstáculos del entorno. El entorno puede estar libre de obstáculos o compuesto de hasta un máximo de tres elementos: cubo, cono (tetraedro) y esfera (octaedro).

A continuación se explica como se detectan las marcas en el entorno de trabajo.

5.1.1. Sistema de referencia

La tarea principal de la biblioteca ARToolKit es obtener la matriz de transformación. La idea es detectar donde se encuentran las marcas y superponer los objetos tridimensionales sobre las mismas. Se aplica la matriz de transformación de la vista para una correcta visualización. Todo ello se realiza en tiempo real, sin retrasos visuales para el usuario. Se debe realizar de forma rápida, ya que se debe realizar la superposición de los objetos tridimensionales.

Antes de continuar con la explicación de la metodología, es importante mostrar como se interpreta el entorno de trabajo (mundo) y su sistema de coordenadas. Este aspecto es de vital importancia, ya que todos los cálculos se realizarán considerando la cámara de la tableta como sistema de referencia.

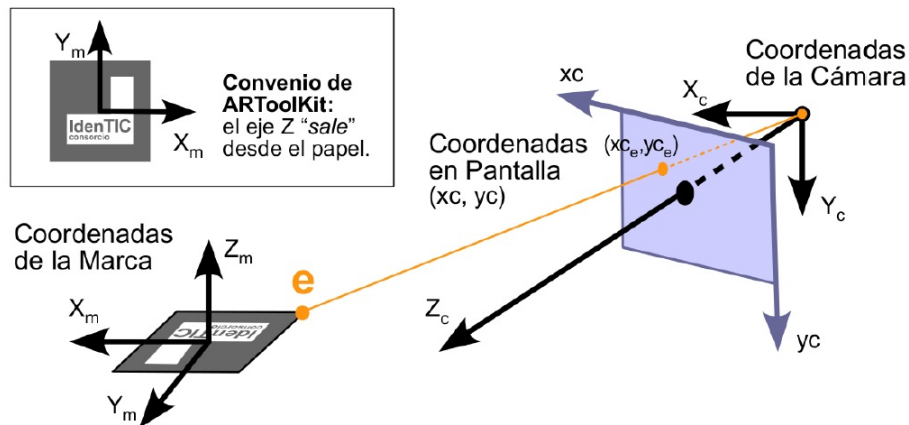


Figura 32: Sistema de referencia

En la figura anterior se pueden observar los principales sistemas de coordenadas del proceso.

- Sistema de coordenadas de la cámara: para el caso del proyecto representa el más importante. Se considera el origen y a partir de él se realizan los cálculos de distancias y control de colisiones.

- Sistema de coordenadas de la marca: se utiliza para saber la posición, colocación y orientación del dibujado en 3D sobre el marcador.
- Sistema de coordenadas de la pantalla: representa las coordenadas de dibujado sobre la pantalla del dispositivo.

Por convenio de ARToolkit, el eje Z del sistema de coordenadas es el que “sale”.

Una vez identificados los elementos de los sistemas de coordenadas de la figura 32, se procede a la obtención de la matriz de transformación.

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

Figura 33: Matriz de transformación

En base a la figura previa, mediante la matriz de transformación se puede obtener la posición (x,y,z) de los marcadores respecto a la cámara y por tanto, conocer la separación en distancia de una marca.

5.1.2. Implementación de visibilidad de los marcadores

A nivel de implementación, la biblioteca ARToolkit incluye métodos para obtener la matriz de transformación. De ella se pueden obtener los valores X-Y-Z y con ellos calcular la distancia de separación entre la cámara y la marca.

```
public void distancia_camara_objeto() {
    double[] matrizTraslacion = this.getTransMatrix();
    double[] centro = {0, 0, 0, 1};
    double x = matrizTraslacion[3];
    double y = matrizTraslacion[7];
    double z = matrizTraslacion[11];

    distancia = Math.sqrt(x*x + y*y + z*z)/1000; //dist en metros
}
```

A parte de conocer las distancias de separación, es muy importante detectar si una marca es visible o no por la cámara. ARToolkit dispone del siguiente método.

```
ObjectSelectorActivity.conoObject.isVisible();
ObjectSelectorActivity.cuboObject.isVisible();
ObjectSelectorActivity.esferaObject.isVisible();
ObjectSelectorActivity.sideObject1.isVisible();
ObjectSelectorActivity.sideObject2.isVisible();
ObjectSelectorActivity.sideObject3.isVisible();
ObjectSelectorActivity.sideObject4.isVisible();
```

Parece algo trivial, pero es de gran importancia detectar si se ha perdido el foco entre la cámara y la marca. De esta forma, sólo cuando las marcas del entorno son visibles, se realizan los cálculos operacionales.

5.2. Detección de colisiones

Una vez se han identificado las marcas y se detectan que son visibles, ya se puede comenzar a tratar los cálculos para controlar las colisiones. A continuación se expone el proceso.

5.2.1. Implementación de detección de colisiones

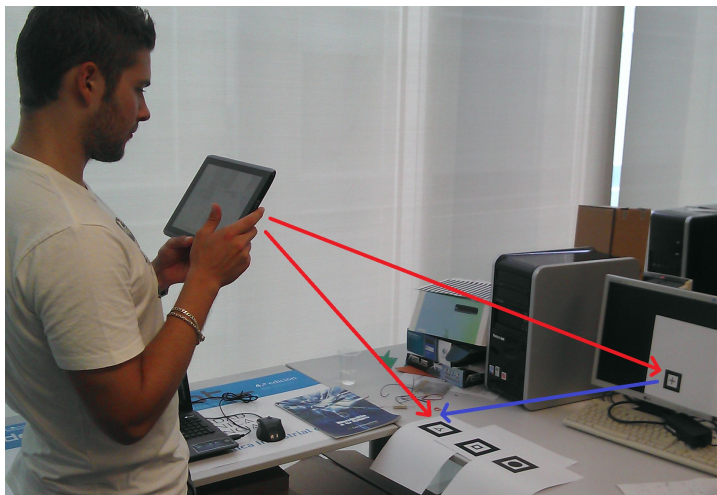


Figura 34: Separación entre cámara y marcas

Dado que se puede obtener la separación entre la cámara y las marcas, se ha integrado una clase “MiVector.java” con la que se pueden realizar operaciones vectoriales. Si se conoce la separación y considerando que la cámara se encuentra en el origen $(0,0,0)$, se puede crear un vector entre la cámara y una marca.

A nivel de implementación se utiliza un bucle dentro de un hilo de ejecución. Dentro del bucle se comprueba constantemente qué objetos son visibles y se traza un vector desde la cámara a cada uno de los objetos. En tiempo real se está determinando la posición relativa de los objetos en función de la cámara (origen).

Una vez se disponen de los vectores de cada una de las marcas visibles, se calcula un nuevo vector separación entre la marca del robot y cada una de las marcas objeto del entorno.

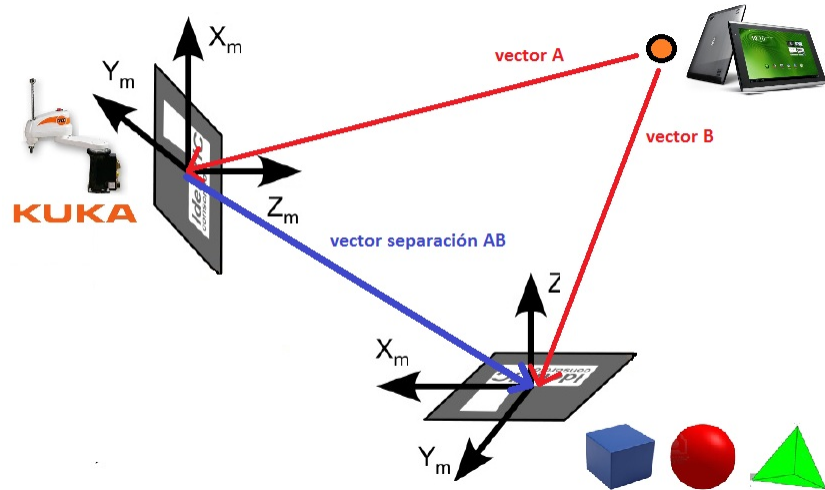


Figura 35: Cálculos vectoriales de los objetos del entorno

En base a la figura anterior, el vector A representa el vector entre la cámara y la marca del robot. Los vectores B, C y D representan los vectores entre la cámara y las marcas que representan los objetos. Estos vectores como tal no son relevantes, los que realmente son importantes son los vectores separación AB, AC y AD respectivamente. A partir de ellos, con obtener el módulo vectorial se determina la separación entre el robot y los obstáculos.

5.2.2. Aproximación por esferas envolventes

En este punto se ha obtenido el cálculo para la separación entre el centro de las marcas. Dado que con la realidad aumentada se representan figuras tridimensionales, se ha de determinar si la colisión se produce con el objeto representado, no con el centro de la marca. Para ello se ha considerado integrar el concepto de esferas envolventes sobre los objetos representados.

La idea de emplear esferas envolventes es poder controlar las colisiones entre los objetos representados por las marcas de una forma eficiente y sin llegar a superponer las marcas entre sí.

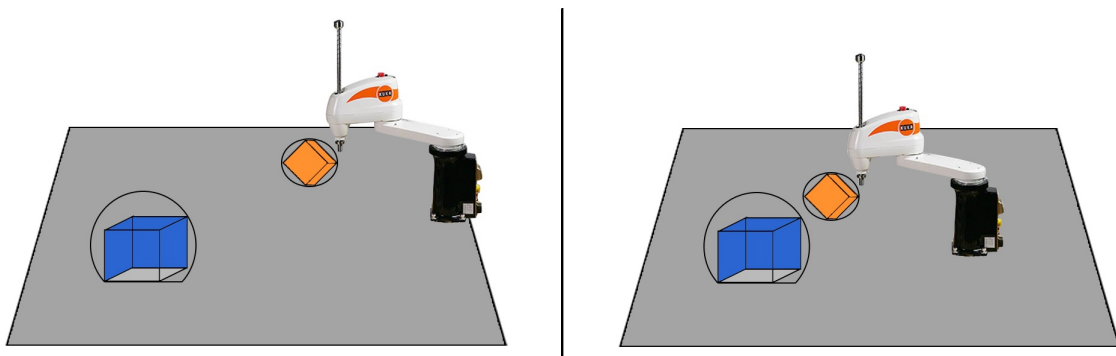


Figura 36: Esferas envolventes

Se ha considerado una esfera envolvente de radio el tamaño de la marca (no importa si se tiene en cuenta el ancho o el largo, ya que son cuadradas). Cada figura 3D representada ha de tener unas dimensiones máximas de la envoltura de la marca. De esta forma al considerar una esfera envolvente de radio de la marca y unas dimensiones máximas de representación del tamaño de la marca, la envolvente siempre abarcará la figura representada.

En resumen, se producirá una colisión entre el robot y otro objeto cuando sus esferas envolventes choquen.

```
double separacion = nuevo.distancia_vector() - radioa - radiob;  
if(separacion <= 0.0) return true;  
else return false;
```

A la hora de realizar los cálculos, existirá riesgo de colisión cuando la separación entre los marcadores, menos los radios de las esferas envolventes sea menor o igual a cero.

5.3. Acción ante colisiones

Una vez se ha identificado si existe riesgo de colisión hay que tomar las medidas oportunas desde la aplicación. Lo primero de todo es indicar al usuario que se está produciendo una colisión entre el robot y uno de los objetos.

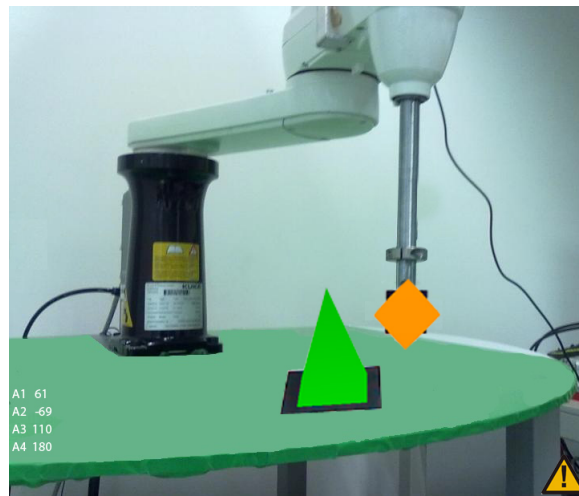


Figura 37: Información sobre una colisión

La forma de hacerlo es mediante la imagen de alarma mostrada en la interfaz de usuario de la aplicación.

Una vez se ha indicado al usuario el riesgo de colisión, se toma la elección de subir el eje del robot a una altura suficiente como medida de seguridad. Además de mandar de forma automática la orden de subir el eje del robot, también se eliminan las otras posibles órdenes posteriores encoladas, ya que pueden ser órdenes de riesgo para futuras colisiones. De esta forma, el eje del robot se queda en alto a la espera de nuevas órdenes de usuario.

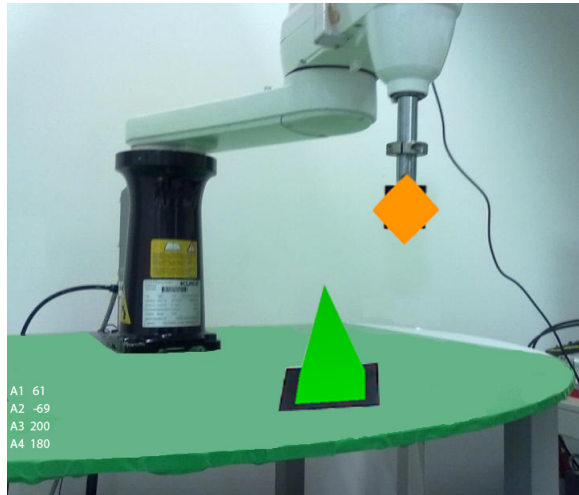


Figura 38: Acción ante una colisión

Para realizar el envío de órdenes desde la aplicación, se ha considerado la misma forma de hacerlo que en el proyecto de Ángel Fernández-Cañada [1]. Se compone una orden (“String”) en el formato adecuado (según lo explicado en la sección 4.2.1) y se envía a la cola de órdenes que el robot va ejecutando. Esta orden tiene un carácter especial, con prioridad, ya que se debe prestar atención al riesgo de colisión antes que al resto de órdenes de usuario.

De la misma forma que se actualizan los movimientos del robot, también se actualiza la interfaz de usuario. Se indican los nuevos valores de los ejes en los que se ha posicionado el robot. Durante el periodo de actualización de movimiento del robot y de interfaz de la aplicación, el usuario puede seguir introduciendo órdenes pero serán encoladas como medida de seguridad. Hasta que el eje del robot no se encuentre en alto y se haya actualizado los nuevos valores de posición de los ejes, no se continuará con los siguientes movimientos de usuario.

5.4. Funcionamiento de la aplicación de colisiones

La aplicación de control de colisiones se han integrado sobre la aplicación original de Ángel Fernández-Cañada [1], es decir, el funcionamiento de la aplicación es la misma pero se han integrado nuevas funcionalidades.

Al arrancar la aplicación principal Kuka Control, aparece un menú de selección. Inicialmente se ha de seleccionar el dispositivo Bluetooth que funcionará como intermediario para el envío de órdenes (es el computador convencional, explicado previamente en la sección 4.2). Por otra parte se puede seleccionar opciones de movimiento del robot (movimientos más rápidos/lentos y otros parámetros de configuración).



Figura 39: Menú de la aplicación Kuka Control

Seguidamente ya se puede arrancar la aplicación de colisiones pulsando el “Modo Táctil”. Si no se introducen marcas de realidad aumentada, el proyecto funciona como el original. Al colocar las marcas sobre el entorno y el robot, de forma automática, se muestra la interpretación de los marcadores y es cuando se pueden realizar los movimientos de usuario considerando las colisiones.

6. Aplicación de selección de objetos

En esta sección se va a mostrar el funcionamiento y desarrollo de la aplicación de selección de objetos, integrada sobre el proyecto de Ángel Fernández-Cañada [1].

En la actualidad existen muchas aplicaciones industriales donde los robots, mediante cámaras y otros sensores, seleccionan objetos y los procesan de forma automática. Con esta aplicación se pretende la misma tarea pero aplicando realidad aumentada.

El objetivo es el siguiente, se colocan las marcas sobre el robot y sobre los objetos del entorno con los que se quiere trabajar. Desde la aplicación y empleando la realidad aumentada, el usuario visualizará los objetos marcados como las figuras mencionadas en los puntos anteriores: cubo, cono (tetraedro) y esfera (octaedro). Automáticamente, la aplicación detectará cuales son los objetos visibles del entorno y los mostrará en la interfaz de usuario. El usuario, de forma táctil, seleccionará el objeto deseado y automáticamente, el robot industrial realizará los movimientos adecuados hasta alcanzar dicho objeto.

6.1. Detección de marcadores

El proceso de identificación de las marcas se basa en el mismo principio que el proceso de identificación de marcas de la aplicación de control de colisiones. Considerando la explicación de la sección 5.1, se detectan cuales son las marcas visibles, se identifican y se muestran sobre la aplicación de usuario mediante realidad aumentada.

6.2. Detección de objetos

Uno de los objetivos de la aplicación es detectar los objetos del entorno que son visibles e indicarlo al usuario. La idea es la siguiente, tanto el robot como los objetos dispondrán de marcas. Siempre y cuando la marca del robot sea visible, la aplicación detectará qué objetos se encuentran en el área de trabajo. Inmediatamente se muestra al usuario cuales son los elementos seleccionables.

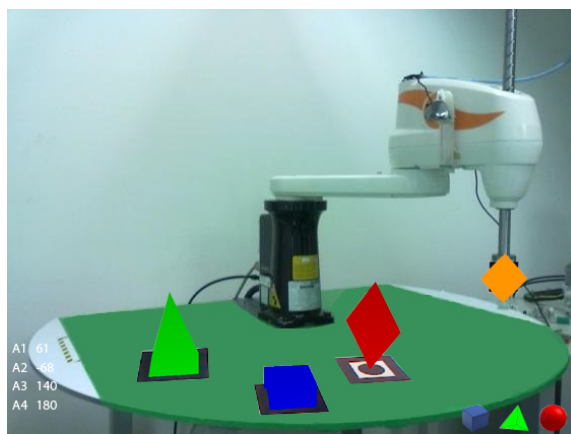


Figura 40: Detección de objetos

Se ha considerado la restricción de que sólo analice los elementos alcanzables del entorno cuando sea visible el robot, de esta forma se evitan movimientos y acciones indeseados por parte del robot. Por otra parte también se ahorra en coste computacional evitando cálculos extra.

6.2.1. Implementación de detección de objetos

Desde el punto de vista del desarrollo e implementación, la aplicación se ejecuta como un bucle dentro de un hilo de ejecución. De forma similar a la sección 5.2.1 de control de colisiones, constantemente

se evalúa qué objetos están presentes en el entorno y mostrándolo por pantalla al usuario.

Al formar parte de un hilo de ejecución independiente permite detectar al instante si aparece o desaparece un nuevo elemento en el entorno.

6.3. Movimiento al objeto

Una vez el usuario puede elegir entre los elementos del área de trabajo y selecciona uno de ellos (cono, cubo o esfera), la aplicación procede a enviar las órdenes adecuadas para que el robot, desde la posición en la que se encuentre, alcance el objeto.

La forma de trabajar es la siguiente. De forma totalmente transparente, se realiza el cálculo vectorial de las posiciones relativas de las marcas, tal y como se muestra en la figura 35. Tras la conocer las posiciones del robot y del objeto seleccionado, se relativizan las posiciones. Se observa, respecto a la referencia de la cámara, si el robot está a la derecha o izquierda del objeto y si está arriba o abajo.

Una vez sabidas las posiciones relativas del robot y del objeto, se procede desde la aplicación, a generar las órdenes adecuadas de movimiento. Es decir, primero, se sube el eje del robot como medida de seguridad. Una vez en alto, con un bucle se van enviando órdenes de movimiento a izquierda o a derecha respectivamente en movimientos de X unidades. Seguidamente, cuando se ha alcanzado la posición adecuada del eje x, con otro bucle, se va moviendo hacia arriba o hacia abajo en movimientos de Y unidades hasta alcanzar la meta. En este punto, se baja el eje del robot a una altura adecuada para tocar el objeto seleccionado.

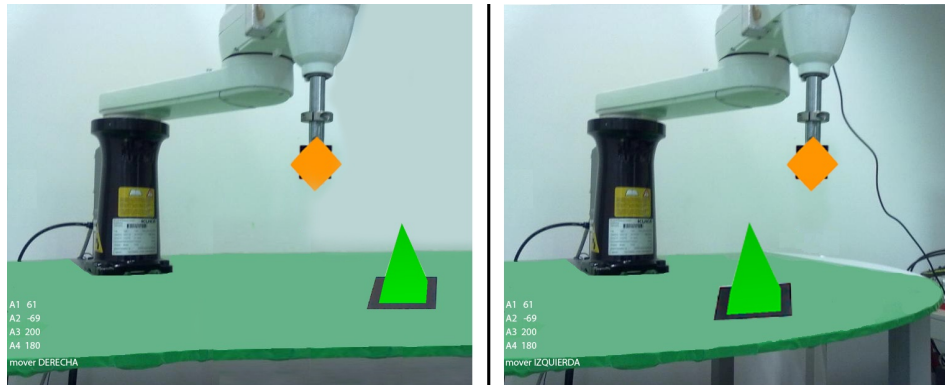


Figura 41: Movimiento al objeto

Tanto en el movimiento del eje x (izquierda o derecha) como el del eje y (arriba o abajo) son fijos. Para evitar que el robot se quede en un bucle moviéndose a un lado y a otro se considera un margen de cercanía. Cuando el robot está lo suficientemente cerca del objeto se detiene y continúa ejecutando instrucciones.

Tras cada cada movimiento se hace un “Sleep” de un segundo aproximadamente para dejar la aplicación bloqueada y dar tiempo a que el robot realice las acciones oportunas.

6.4. Funcionamiento de la aplicación de selección de objetos

Las aplicación de selección de objetos se ha integrado sobre la aplicación original de Ángel Fernández-Cañada [1], como un menú nuevo seleccionable.

Al arrancar la aplicación principal Kuka Control, aparece un menú de selección (figura 39). Inicialmente se ha de seleccionar el dispositivo Bluetooth que funcionará como intermediario para el envío de órdenes (es el computador convencional, explicado previamente en la sección 4.2). Por otra

parte se puede seleccionar opciones de movimiento del robot (movimientos más rápidos/lentos y otros parámetros de configuración).

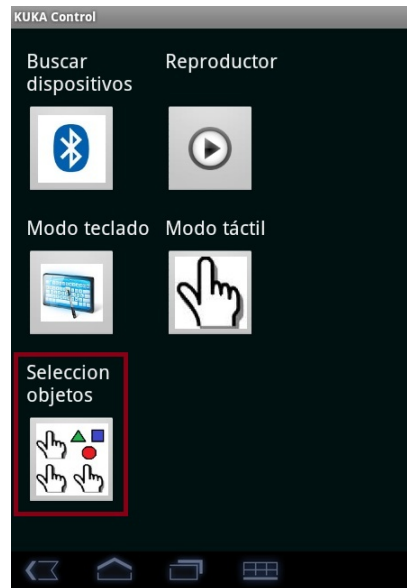


Figura 42: Selección de la aplicación en el menú de Kuka Control

Seguidamente ya se puede arrancar la aplicación de selección de objetos pulsando el botón de “Selección de Objetos”. Si no se introducen marcas de realidad aumentada, el proyecto funciona “en abierto”, es decir, el usuario puede hacer que el robot se mueva a petición suya sin restricciones. Al colocar las marcas sobre el robot y el entorno, automáticamente se detectan y permite la opción de seleccionar el objeto deseado a alcanzar.

7. Conclusiones

En este proyecto se ha podido aplicar la técnica de realidad aumentada como un elemento de sensorización. A nivel industrial, esta técnica se utiliza habitualmente para mostrar información en pantalla sobre los elementos que poseen marcadores. En el proyecto se ha mostrado la funcionalidad de la realidad aumentada y las posibles aplicaciones sobre las que se puede trabajar.

La realidad aumentada es un sistema que cada vez más se está desarrollando y aplicando. Es práctico, sencillo y barato (basta con marcas de papel y cualquier dispositivo que las interprete). En el proyecto se ha expuesto que es un sistema fiable y seguro, que tiene una gran funcionalidad y que se puede aplicar en proyectos industriales futuros.

El enfoque más importante quizá sea precio. Para trabajar con la realidad aumentada no es necesario más que elementos que todos los usuarios disponen en la vida diaria. A nivel industrial y dependiendo de las aplicaciones sobre las que se quiera aplicar, se reduciría el coste considerablemente de cambiar los sensores habituales por la integración de realidad aumentada. Es un sistema intuitivo y que permite la interacción con el usuario si se desea.

Para desarrollar aplicaciones con realidad aumentada no es necesario conocimientos especiales salvo los propios que requiera el dispositivo. La mayoría de dispositivos móviles utilizan sistema operativo Android, iOS y Symbian. El desarrollo de aplicaciones sobre estas plataformas está a la orden del día y existe mucha documentación al respecto.

Cada vez más se desarrollan nuevas librerías adaptadas para realidad aumentada. Dependiendo de sus aplicaciones: reconocimiento de caras, reconocimiento de formas... existen librerías específicas para trabajar con las tareas necesarias.

Se ha podido comprobar el nivel de seguridad que puede proporcionar la realidad aumentada, incluso en un campo tan delicado como la seguridad ante obstáculos. Es cierto que dependiendo de la complejidad del sistema sobre el que se quiera aplicar la realidad aumentada, la fluidez y rapidez de las aplicaciones puede variar. En el proyecto se ha visto como la información que introduce el usuario pasa por varios intermediarios hasta alcanzar el robot con soltura, sin embargo, cada entorno de trabajo es diferente y es necesario ver si la realidad aumentada es el sistema adecuado.

Cada sensor industrial tiene sus limitaciones y restricciones. En el caso de la realidad aumentada, como se ha explicado, tiene otros factores que limitan su funcionamiento. Se requieren espacios de trabajo con luz, donde la separación del área de trabajo y la cámara del dispositivo sea relativamente pequeña (relación entre distancia de separación de la cámara y tamaño de las marcas).

En general la realidad aumentada es un sistema fiable y robusto en continuo desarrollo y que cada vez más se aplica en múltiples aplicaciones de la vida diaria.

8. Trabajo Futuro

En el proyecto se ha explicado el funcionamiento de las aplicaciones de realidad aumentada sobre el robot industrial SCARA y el entorno de trabajo. Sobre ello hay ciertos aspectos que se podrían considerar como desarrollo de trabajo futuro.

Como se ha explicado en el proyecto (en la sección 4.2) la información pasa por una serie de etapas y codificaciones de órdenes. Primero el usuario introduce las órdenes al dispositivo móvil, seguidamente y de forma transparente, se envían al computador convencional vía Bluetooth. Tras realizar la codificación e interpretación de las órdenes, se envían al servidor del robot por un cable de red. Por último, el servidor ya envía las órdenes apropiadas al robot.

La idea es poder introducir una tarjeta de red inalámbrica sobre la unidad de control. De esta forma todo el proceso de codificación de órdenes se haría de forma transparente desde la aplicación principal del dispositivo móvil. Seguidamente se enviarían las órdenes directamente por Wi-Fi desde el dispositivo móvil a la unidad de control del robot. Así se eliminaría el intermediario del computador convencional, evitando posibles pérdidas de información y obteniendo una considerable fluidez sobre los movimientos del robot.

Tanto en la aplicación de “Control de Colisiones” como en la de “Selección de Objetos” se trabaja con tres posibles objetos de realidad aumentada: un cubo, un cono (tetraedro) y una esfera (octaedro). Las figuras 3D que se representan de las marcas, se introducen a nivel de código Java. Es cierto, que no es complicado añadir nuevas figuras replicando código, pero sería interesante implementar una aplicación con la que se puedan introducir nuevos elementos y configurarlos.

Por otra parte, en el proyecto se ha explicado que las figuras que se representan tridimensionalmente a partir de las marcas, están constituidas por triángulos, y por tanto, son figuras “sencillas” por ahorro de cómputo. Sería interesante poder llegar a un equilibrio entre la complejidad de las figuras representadas y la eficiencia de los cálculos de control. El hecho de poder interpretar figuras 3D complejas, puede hacer que la aplicación sea más intuitiva y atractiva para el usuario.

Habitualmente en la industria, la realidad aumentada se utiliza como paneles de información. Al enfocar con el dispositivo sobre la marca posicionada en un determinado lugar, el usuario obtiene información por pantalla del objeto o sistema. Además de las aplicaciones desarrolladas en el proyecto, sería interesante añadir una nueva funcionalidad para obtener y visualizar información del robot. Enriquecería la aplicación principal con mayor funcionalidad.

9. Bibliografía

- [1] PFC de Ángel Fernández-Cañada Vilata (Septiembre 2012)
Programación de un robot industria scara mediante un dispositivo móvil
- [2] PFC de Pablo Muñoz Rodríguez (Septiembre 2011)
Desarrollo de aplicaciones integrando robótica y visión en un robot industrial KUKA para demostrar sus capacidades
<http://riunet.upv.es/bitstream/handle/10251/11908/memoria.pdf?sequence=1>
- [3] PFC de Carlos Alcarria Izquierdo (2010)
Desarrollo de un sistema de Realidad Aumentada en dispositivos móviles
- [4] PFC de César Mora Castro (Julio 2011)
Minerva: Sistema de especificación lógica basado en Sensores, Controladores y Actuadores para aplicaciones de Realidad Aumentada
<http://www.inf-cr.uclm.es/www/cglez/downloads/pfc/pfccmora.pdf>
- [5] PFC de Ricardo Muñoz Castellanos (Febrero 2012)
ARGES: Sistema de Posicionamiento en Interiores Basado en Técnicas de Realidad Aumentada
<http://www.esi.uclm.es/www/cglez/downloads/pfc/pfcrmunoz.pdf>
- [6] Paul Milgram y Fumio Kishino (1994)
A Taxonomy of Mixed Reality Visual Displays
- [7] Ronald T. Azuma (1997)
A Survey of Augmented Reality
<http://www.cs.unc.edu/~azuma/ARpresence.pdf>
- [8] Douglas Holmgren (1992)
Design and Construction of a 30-Degree See-Through Head-Mounted Display
<http://www.cs.unc.edu/techreports/92-030.pdf>
- [9] Ivan Sutherland (1968)
A Head-Mounted Three Dimensional Display, pp. 757-764.
<http://141.84.8.93/lehre/ss09/ar/p757-sutherland.pdf>
- [10] T.P.Caudell y D. W. Mizell (1992)
Augmented Reality: An Application of Heads-Up Display Technology to Manual Manufacturing Processes, pp. 659-669.
- [11] H. Kato y M. Billinghurst (1999)
Marker tracking and HMD calibration for a video-based augmented reality conferencing system, pp. 85-94.
http://cgkit.nutn.edu.tw:8080/cgit/PaperDL/CUC_110708153503.PDF
- [12] S. Julier, Y. Baillot, M. Lanzagorta, D. Brown y L. Rosenblum (2000)
BARS: Battlefield Augmented Reality System
<http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADP010892>
- [13] M. Kalkusch, T. Lidy, M. Knapp, G. Reitmayr, H. Kaufmann y D. Schmalstieg (2002)
Structured Visual Markers for Indoor Pathfinding, pp. 8

- [14] A. Henrysson, M. Billinghurst, and M. Ollila (2005)
Face to Face Collaborative AR on Mobile Phones, pp. 80-89.
- [15] G. Reitmayr, and D. Schmalstieg (2001)
Mobile Collaborative Augmented Reality, pp. 114-123.
<http://www.ims.tuwien.ac.at/media/documents/publications/mobileISAR2001.pdf>
- [16] Información sobre KUKA
<http://es.wikipedia.org/wiki/KUKA>
Consultado en agosto del 2012
- [17] Información sobre el robot SCARA (KUKA)
<http://www.kuka-robotics.com/es/>
Consultado en agosto del 2012
- [18] Definición y conceptos de la realidad aumentada
http://es.wikipedia.org/wiki/Realidad_aumentada
<http://www.maestrosdelweb.com/editorial/que-es-realidad-aumentada/>
Consultado en agosto del 2012
- [19] Esferas envolventes
Apuntes de Robótica Industrial Avanzada.
Máster en Automática e Informática Industrial 2011-2012. Universidad Politécnica de Valencia.
Consultado en abril del 2012
- [20] Librerías de realidad aumentada
<http://ingcarlosreina.inkframe.com/realidad-aumentada/librerias-de-realidad-aumentada/>
Consultado en febrero del 2012
- [21] Documentación sobre AndAR
<http://code.google.com/p/andar/>
Consultado en febrero del 2012
- [22] Identificación de marcas de realidad aumentada
http://integrar.bue.edu.ar/wp-content/uploads/2011/09/Tutorial_Marcadores_Realidad_Aumentada.pdf
<http://code.google.com/p/andar/wiki/HowToUseYourOwnMarkers>
Consultado en febrero del 2012
- [23] Sistema de referencia en realidad aumentada
http://www.esi.uclm.es/www/cglez/realidad_aumentada/downloads/transparencias/sesion01_02.pdf
Consultado en febrero del 2012
- [24] Información sobre la tableta del entorno de trabajo
<http://www.acer.es/ac/es/ES/content/iconia-tab-a500>
<http://www.muycomputer.com/2011/05/10/acer-iconia-a500-32-gbytes>
Consultado en febrero del 2012
- [25] Información sobre KRC Editor
<http://www.krceeditor.de/blog/>
Consultado en diciembre del 2011

[26] Información sobre dispositivos portátiles
<http://www.wikipedia.org>
Consultado en diciembre del 2011

[27] Información sobre campos de aplicación
<http://www.wikipedia.org>
Consultado en diciembre del 2011

[28] Información sobre Wikitude
<http://www.wikitude.com/>
Consultado en agosto del 2012

