

**UNIVERSIDAD DE SALAMANCA**  
**FACULTAD DE CIENCIAS**



**INGENIERÍA TÉCNICA EN**  
**INFORMÁTICA DE SISTEMAS**

**TÍTULO**

**TELEOPERACIÓN DE UN BRAZO**  
**ROBÓTICO MEDIANTE REALIDAD**  
**AUMENTADA**

**AUTOR**

**D. ALBERTO RODRÍGUEZ VALLE**

**TUTORES**

**D. ROBERTO THERÓN SÁNCHEZ**

**DÑA. BELÉN CURTO DIEGO**



## **1 Introducción**

En la última década la convergencia de capacidades de procesamiento de imágenes de video en tiempo real, sistemas gráficos por ordenador y nuevas tecnologías de visualización han hecho posible la obtención de una imagen virtual correctamente insertada con la visión del entorno tridimensional que rodea al usuario. Un sistema de Realidad Aumentada proporciona una visión compuesta para el usuario: se trata del resultado de la combinación de una escena real y otra virtual generada por el ordenador que complementa la primera con información adicional. El objetivo final es crear un sistema en el que el observador no consiga discernir la diferencia entre el mundo real y la versión aumentada del mismo. El usuario de este sistema tendría la sensación de observar una única escena real.

La investigación entorno a la Realidad Aumentada se centra fundamentalmente en dos aspectos: por una parte el desarrollo de métodos para insertar y mantener las dos imágenes en tiempo real, y por otra, la fusión de ambas imágenes.

Los investigadores que trabajan con sistemas de Realidad Aumentada los utilizan en diversos dominios como son: la medicina, el entretenimiento, la televisión, la robótica, el diseño orientado al consumidor, el diseño de ingeniería, mantenimiento de equipos, etc.

Este proyecto lleva la Realidad Aumentada al mundo de la robótica en una vertiente didáctica y otra práctica.

Desde la perspectiva didáctica, la Realidad Aumentada se aplicará para facilitar la comprensión del método de Denavit y Hartenberg a un robot tipo PUMA que se encuentra en el Laboratorio de Robótica del Departamento de Informática y Automática. Este método se utiliza de forma muy intensa para resolver el problema de la cinemática de un robot. Así el alumno podrá comprender cómo se posiciona los ejes de los sistemas de referencia de Denavit y Hartenberg en las articulaciones del robot y como varían los parámetros de este método.

## **2 Objetivos**

En este trabajo se estudia una biblioteca de Realidad Aumentada y se emplea en el desarrollo e implementación de una aplicación que conduce el movimiento de un brazo robótico. En consecuencia, para el desarrollo de este proyecto se proponen los siguientes objetivos:

- Estudiar y valorar, una biblioteca de Realidad Aumentada.
- Explorar las posibilidades que ofrece la Realidad Aumentada en el desarrollo de aplicaciones.
- Conducir un brazo robótico hasta un objeto tangible, utilizando patrones reconocibles.
- Diseñar una aplicación que muestra de forma visible e intuitiva la variación de los parámetros de Denavit-Hartenberg en un brazo robótico de seis grados de libertad.
- Estudiar el rendimiento de la biblioteca y de la dos aplicaciones desarrolladas mediante *profiling*.

## **3 Aspectos relevantes**

Existen diferentes herramientas que facilitan la inserción de imágenes virtuales dentro del mundo real. Entre ellas cabe señalar las bibliotecas ARToolkit, ARToolkitPlus y JARToolkit, que utilizan patrones para determinar la posición en la que se desea insertar el objeto virtual. Estos patrones son cuadrados con un borde negro que contienen una figura en su interior.

Después de analizar los resultados de las pruebas del reconocimiento de patrones realizadas en las mismas condiciones con las tres bibliotecas, se optó por la biblioteca ARToolkit para el desarrollo de este proyecto. La decisión se basó en su mayor fiabilidad y en que posee herramientas para calibrar las cámaras y crear nuevos patrones. Las otras bibliotecas carecen de estas herramientas que facilitan el desarrollo de las aplicaciones.

Esta biblioteca convierte las imágenes que recibe de una webCam a una escala de grises y detecta los cuadrados que hay en cada imagen y los compara

con los que almacena en su memoria. Una vez reconocidos, proporciona su posición en un sistema de coordenadas centrado en la cámara mediante una matriz de transformación.

En este proyecto se han empleado las técnicas de la Realidad Aumentada para insertar los ejes de distintos sistemas de referencia sobre las imágenes de las articulaciones de del brazo robótico MA2000 captadas por webCams con una doble finalidad: determinar la configuración del brazo robótico y facilitar la comprensión del posicionamiento de los ejes del método de Denavit y Hartenberg. Este método consiste en fijar unos sistemas de referencia a partir de los cuales se obtienen los parámetros asociados a cada articulación para determinar la posición en la que se encuentra el extremo del robot.

El robot MA2000 dispone de seis grados de libertad, en consecuencia, para determinar la configuración del robot y la posición en la que deben situarse los ejes se utiliza un patrón por cada grado de libertad y uno más para la base. Estos patrones se colocan en cada una de las articulaciones.

La obtención de imágenes se realizó inicialmente mediante una única cámara, pero con la aparición de ángulos muertos para la cámara ocasionados por el movimiento del brazo, se optó, en primer lugar, por incrementar a dos el número de cámaras y, finalmente a tres. De esta forma se consiguió ampliar el campo de observación.

Cada cámara debe estar conectada a un puerto *usb* raíz distinto, ya que, en otro caso, no se reciben las imágenes de alguna de las cámaras.

La biblioteca ARToolkit permite todas las cámaras que acepte el sistema operativo, pero, al utilizar muchas variables estáticas, mezcla los patrones que detectan las distintas cámaras. Por ello se ha creado un proceso independiente que controla a cada cámara, y un proceso más que realiza los cálculos y maneja la interfaz.

Para comunicar los procesos se ha utilizado una zona de memoria compartida y un semáforo para cada cámara.

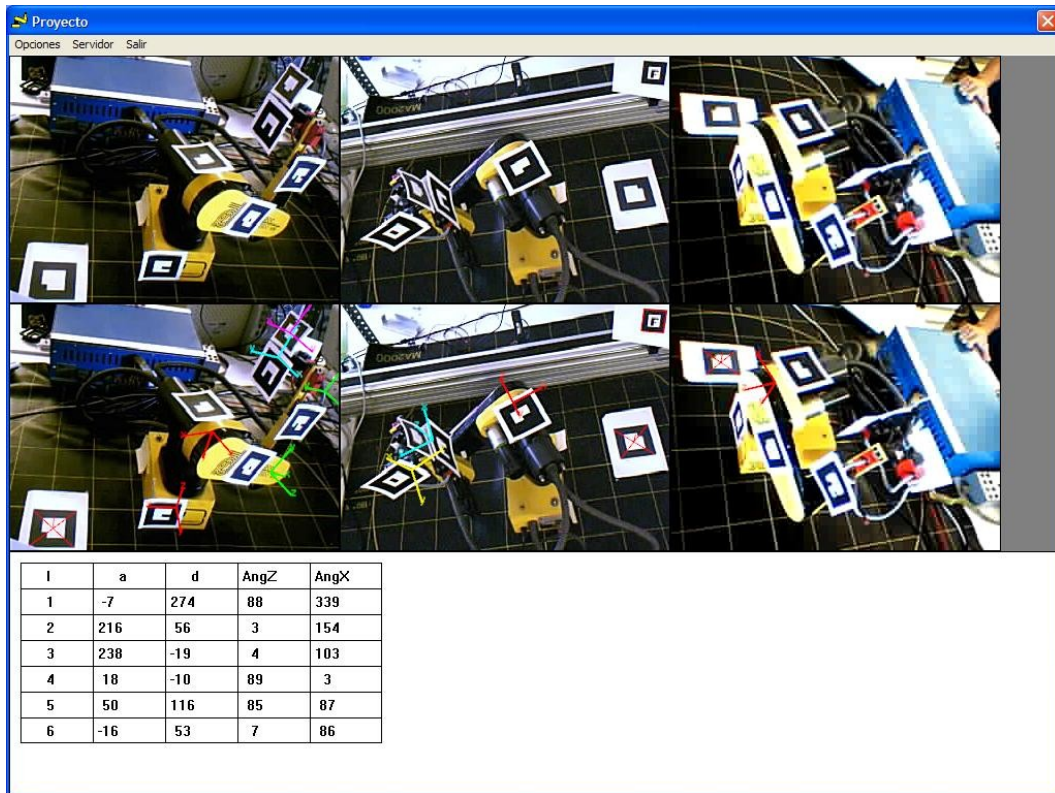
Para conseguir la operatividad de los datos facilitados en sistemas de referencia distintos, se han transformado todos ellos a un nuevo sistema de referencia situado en un patrón visible a las tres cámaras al mismo tiempo. Una vez obtenidas todas las posiciones de todos los patrones en un mismo sistema de referencia, la determinación de las distancias y los ángulos entre las distintas articulaciones se realizan mediante cálculos vectoriales.

El método de Denavit y Hartenberg establece de manera sistemática un sistema de coordenadas ( $F_i$ ) ligado a cada eslabón “i” de una cadena articulada, permitiendo determinar a continuación las ecuaciones cinemáticas de la cadena completa.

A partir de la representación de Denavit y Hartenberg se pasa desde el sistema de referencia asociado a cada eslabón al sistema del siguiente mediante cuatro transformaciones básicas que dependen exclusivamente de las características geométricas del eslabón.

Estas transformaciones básicas consisten en una sucesión de rotaciones y traslaciones que permitan relacionar el sistema de referencia del elemento “i” con el sistema de elemento “i-1”. Estos sistemas de coordenadas se representan sobre cada una de las articulaciones del robot MA2000 en diferentes colores para su mejor visualización.

Para completar la información sobre el método de Denavit y Hartenberg se muestra en pantalla la tabla de parámetros que reflejan las características y la configuración del robot.



**Figura 1: Captura de la aplicación**

La posición en la que debe situarse el extremo del brazo robótico se indica mediante un nuevo patrón. Sus coordenadas se obtienen respecto del sistema referencia de la base del robot.

Para desplazar el brazo desde la configuración en la que se encuentra hasta la posición objetivo determinada por el patrón se utiliza Player.

Player es un servidor multihilo de dispositivos robóticos que se ejecuta en cualquier plataforma tipo UNIX. Proporciona a los programas clientes un control simple y completo sobre los dispositivos sensores y actuadores a través de la red. Se puede ejecutar bien en el propio robot o en un ordenador al que estén conectados los dispositivos que se desean manejar. Soporta tanto conexiones serie como TCP/ IP, permite el control de múltiples robots simultáneamente y la integración de nuevos dispositivos para ser controlados a través de él.

Para el desarrollo de este proyecto se ha utilizado la ampliación de Player llevada a cabo por el Departamento de Informática y Automática de la Universidad de Salamanca para el robot MA2000. Esta ampliación inserta un

*driver* para cada uno de los dispositivos del robot y una interfaz mediante la cual el usuario se comunique con él.

En este proyecto, para comunicar la posición del patrón objetivo, que se calcula con la aplicación que se ejecuta en Windows, con el cliente de Player, que se ejecuta en Linux, se optó por diseñar un *driver* genérico que añade una capa más a la arquitectura cliente servidor. De esta manera, se ejecuta una aplicación servidor que se encarga de acceder a los dispositivos, configuración y posición objetivo a petición de Player, que hace de intermediario entre los programas clientes y la aplicación servidora.

El *driver* creado se conecta con el servidor que se encuentra en la aplicación de Windows y, a través de un protocolo de comunicación propio, envía los datos de la configuración del robot y de la posición del objetivo a Player.

Una vez que el cliente de Player recibe la posición del objetivo, se calcula mediante la cinemática inversa la configuración que debe adquirir el brazo y se la comunica al *driver* del robot.

## **4 Conclusiones**

Ha transcurrido poco más de una década desde que aparecieron las primeras técnicas de Realidad Aumentada y, en este período de tiempo, tanto la diversidad de sectores en los que se ha aplicado como el número de proyectos que se han desarrollado ha crecido progresivamente. De la información obtenida en los trabajos consultados para la realización de este proyecto se ha podido comprobar que la Realidad Aumentada utiliza medios informáticos para sobreponer escenas virtuales sobre escenas reales originando en el observador la percepción, en tiempo real, de una escena única, marcando así su diferencia con la realidad virtual.

Existen diversas bibliotecas en diferentes lenguajes y entornos que facilitan el desarrollo de aplicaciones en este campo. Así, en este proyecto se han evaluado diferentes bibliotecas, como son: ARToolkit, ARToolkitPlus, JARtoolkit escritas con distintos lenguajes de programación, para el desarrollo de aplicaciones donde se utiliza la Realidad Aumentada.



Después de realizar esta valoración se ha elegido ARToolkit. Esta biblioteca está diseñada fundamentalmente para la superposición de objetos virtuales en un flujo de imágenes y en este campo proporciona unos excelentes resultados, pues presenta oscilaciones mínimas en la representación de las escenas superpuestas. En cuanto a la determinación de distancias y ángulos proporciona unos resultados de una fiabilidad aceptable. Las desviaciones observadas, que se hemos analizado en este proyecto, se deben principalmente a la calibración de las cámaras, a los reflejos y sombras sobre los patrones. Se obtendrán mejores resultados cuanto mejor sea la calibración de las cámaras y las condiciones de iluminación. Además, se ha comprobado que la biblioteca ARToolkit reconoce mejor los patrones cuanto mejor sea su contraste con el medio.

Sin embargo, ARToolkit presenta algunas limitaciones. Así, los patrones que se van a reconocer deben cumplir unos determinados requisitos en cuanto a su geometría, el marco externo, etc. Además, la biblioteca GLUT para la representación de imágenes por pantalla, asociada a la biblioteca ARToolkit, no dispone de herramientas para el diseño de interfaces atractivas para el usuario, por lo que es conveniente utilizar otras bibliotecas. Para resolver este inconveniente, en este proyecto se ha utilizado la biblioteca SDK de Windows que proporciona una gama de elementos muy variada para diseñar una interfaz gráfica de usuario con unas características atractivas. Esto nos ha llevado a tener que adecuar el dispositivo HDC al uso de primitivas de OpenGL, modificando su configuración, renombrando determinadas funciones y alterando el identificador del formato de la imagen RGBA para OpenGL bajo SDK.

También esta biblioteca presenta algunos problemas cuando se maneja más de una webCam en una misma aplicación, debido a que en la biblioteca se utilizan muchas variables estáticas. Este condicionante ha sido necesario tenerlo en cuenta en la fase de diseño.

En este proyecto se han realizado dos aplicaciones, una con una perspectiva didáctica y la otra con una perspectiva práctica, que llevan la Realidad Aumentada al mundo de la Robótica. Como escenario de pruebas se ha dispuesto del robot MA2000, un brazo robótico tipo PUMA con seis grados de libertad, y de

tres webCams colocadas según una estructura cúbica para obtener un amplio campo de observación. Las imágenes capturadas aparecen simultáneamente en pantalla en su versión real y en su versión de Realidad Aumentada

En la primera se ha introducido la Realidad Aumentada para posicionar los sistemas de referencia del método de Denavit-Hartenberg sobre la imagen de un robot real, en cada una de sus seis articulaciones. De esta forma los alumnos podrán comprender el método de Denavit y Hartenberg, método muy difundido en el ámbito de la Robótica para obtener la cinemática de un robot. Para ello, se ha colocado un patrón diferente sobre cada articulación, y en la imagen de Realidad Aumentada se ha reemplazado por el sistema de referencia correspondiente. También se muestran de forma numérica los parámetros de Denavit y Hartenberg que determinan la configuración del brazo.

En la segunda se ha utilizado la Realidad Aumentada para mover de forma automática un manipulador real desde una configuración actual a otra objetivo, identificada mediante un patrón.

En ambos casos la biblioteca ARToolkit ha servido para detectar e identificar los patrones y calcular su posición y orientación respecto del sistema de referencia de la cámara. Estos patrones se han utilizado para identificar y calcular las seis articulaciones del robot y el punto de destino que debe alcanzar.

Sin embargo, como se utilizan tres *webcams* para lograr un amplio campo de visión, aparece el problema de calcular las distancias y los ángulos entre dos patrones reconocidos por dos cámaras distintas. Así, hemos planteado en nuestro proyecto utilizar de un patrón de referencia, para calcular las matrices de transformación de todos los patrones en relación de él.

Ante la imposibilidad práctica de situar los patrones en los centros geométricos de las articulaciones ha sido necesario modificar las matrices de transformación mediante traslaciones que han aproximado los ejes a dichos centros. Se ha considerado que los cálculos realizados han conllevado un margen de error aceptable, tal y como se ha expuesto el estudio presentado en la memoria de este Proyecto.

La inserción de esta aplicación como un *driver* de Player ha permitido usarla sin modificación alguna como un medio para conocer fácilmente la configuración del robot y obtener la posición del patrón objetivo. Esto ha facilitado conducir el brazo robótico hasta la posición de destino, ya que mediante Player se puede controlar el movimiento del brazo.

Las **líneas futuras de trabajo** basadas en este proyecto pueden ir ligadas a aumentar la interacción entre el usuario y la aplicación, posibilitando la elección del lugar de posicionamiento del brazo no solo mediante un patrón, sino que también se pueda elegir un punto común a las tres cámaras y pinchando sobre cada una de las ventanas asociadas a cada cámara se determine la posición de ese punto objetivo, respecto de la base del robot mediante triangulación.

Otro aspecto interesante será desarrollar una ampliación de la aplicación que permita al brazo robótico salvar los obstáculos, marcados mediante patrones, que se encuentren en su trayectoria.