

DISEÑO DE UN AMBIENTE VIRTUAL MEDIANTE VISIÓN ARTIFICIAL PARA APLICACIONES EN TELEROBÓTICA

Jaime Masiá
jmasia@mcm.upv.es

Juan Ramón Rufino
jrufino@eln.upv.es

Francisco Javier Tormo
fratorg7@epsa.upv.es

José Ferri
joferse2@epsa.upv.es

Borja Mansilla
bormanlv@epsa.upv.es

Universidad Politécnica de Valencia, Campus De Alcoi. Plaza Ferrándiz y Carbonell s/n, 03801 Alcoy (Alicante)

Resumen

Este trabajo pretende dar a conocer una forma alternativa de controlar, teleoperar y programar robots industriales apoyándose en una comunicación mediante técnicas de visión artificial. Se pretende mejorar la interacción hombre-máquina a través de un interface más sencillo e intuitivo.

Esta comunicación entre hombre y maquina se consigue capturando los movimientos de la mano y trasladando estos al manipulador. Dada la similitud morfológica entre el brazo y el manipulador, el uso será más intuitivo que la teleoperación convencional mediante periféricos como joysticks, teclados, palancas, etc.

Para el caso que nos ocupa hemos elegido un robot con la configuración SCARA de 3 grados de libertad. Para el control y generación de la señal PWM necesaria para el posicionamiento de los servomotores se ha utilizado un μC para el diseño hardware. La solución del problema cinemático inverso del robot ha sido implementada en lenguaje C al igual que el interface creado para la ocasión, donde se ha utilizado la librería de código abierto "OpenCV". En la cual utilizando un patrón de referencia determinaremos la posición en el que se evaluarán las componentes X e Y para el posicionamiento del robot en el plano y la componente Z en función del tamaño de este.

El alcance de la técnica abarca desde la teleoperación de robots industriales, robots médicos

o para ambientes hostiles, programación/enseñanza de robots industriales, etc.

Palabras Clave: Ambientes virtuales, telerobotica, visión artificial, opencv.

1 INTRODUCCIÓN

Mediante las actuales técnicas de teleoperación, telerobotica y telepresencia se pueden obtener infinidad de modelos de interacción hombre-máquina basadas en múltiples periféricos o elementos, la visión artificial es una más dentro de este amplio abanico de posibilidades. Este trabajo tiene como fin estudiar la visión artificial como sistema de teleoperación e intentar sentar la base para futuros desarrollos y metodologías.

El consumo energético, la robótica y la mejor integración de las nuevas tecnologías en la sociedad, temas de candente actualidad, unos por sus avances y otros por sus sequías venideras, posibilitan/favorecen el desarrollo de nuevos sistemas que permitan una mejor interacción hombre-máquina. La virtualización de muchos elementos mecánicos, la cibernetización virtual de operarios, periféricos virtuales, conceptos que parecen deshumanizar a la sociedad, no son sino una pequeña solución (un aporte en pro de la solución) para algunos de los problemas de nuestros días, el consumo energético.

Mediante esta técnica podemos obtener una serie de ventajas de carácter técnico, anteriormente mencionadas y energéticas, desde el punto de vista del consumo generado por la fabricación de elementos existentes en los actuales interfaces y el generado por el mantenimiento de estos en las estaciones de teleoperación. La eliminación de estos puede ser un punto a favor de la implantación de esta técnica.

El inconveniente encontrado es el riesgo y la incertidumbre que genera la implementación de una nueva forma de trabajo o técnica debido a la no amortización del sistema empleado en la industria donde se implantará esta nueva metodología de trabajo.

2 DEFINICIÓN DEL ENTORNO

Entendemos como ambientes virtuales inmersivos o no inmersivos [3], [4], como aquellos interfaces que sumergen al usuario en un mundo virtual y que de forma intuitiva facilitan la comunicación o la interacción entre hombre y máquina, requiriéndose o no realimentación sensorial.

Bajo este concepto, la adaptación de los ambientes virtuales puede verse deformada según el sistema en cuestión.

La opción elegida está a medio camino ya que aunque no existe una inmersión total desde el punto de vista panorámico del usuario, función casi elemental para ser considerado como ambiente virtual inmersivo, la forma de interactuar si podría estar dentro de la definición de ambiente virtual inmersivo, ya que los movimientos interpretados por el usuario son de carácter natural (con lo que esto ayuda a una mejor inmersión en el ambiente virtual).

Principalmente los ambientes virtuales, más en concreto los inmersivos, conocidos como realidad virtual, tenían en el campo multimedia y de la simulación su campo de incidencia, también en el diseño de elementos arquitectónicos, y hoy en día los no inmersivos dentro de las redes sociales tienen en ellas su campo de mayor impacto. Aunque en nuestro caso simplemente adaptamos el concepto y lo introducimos en el campo industrial, ya que las ventajas técnicas ofrecidas cubren un amplio rango de tareas.

3 INTERFAZ HOMBRE-MÁQUINA BASADA EN VISIÓN ARTIFICIAL

De este modo se consigue uno de los objetivos principales de este artículo, la creación de interfaces

más sencillos e intuitivos, debido a la anulación de los movimientos antinaturales realizados o descritos mediante los periféricos habituales. Dada la versatilidad y la potencia de la visión por computador estos interfaces son “fáciles” de crear gracias al desarrollo de librerías como OpenCV. Esto junto a la potencia y velocidad de procesamiento de los actuales pc's, tablets, etc. los interfaces pueden ser implementados en cualquier industria o sector.

La metodología de trabajo será el reconocimiento del patrón (mano del operario) con el consiguiente cálculo de su posición relativa al eje deseado y posicionar el manipulador en esta posición.

3.1 IMPLEMENTACIÓN DEL SISTEMA DE VISIÓN ARTIFICIAL

Como se ha comentado, para implementar el sistema de visión artificial se emplea la librería OPENCV [2]. Se considera de interés comentar la función utilizada para detectar patrones.

HaarDetectObjects [2] esta función utiliza la técnica desarrollada por Paul Viola y Michael Jones (detección Viola-Jones [5]) posteriormente Rainer Lienhart y Jochen Maydt [6] continuaron el desarrollo por características diagonales. En OpenCV se denominan “Haar Classifier”, en concreto Haar-like wavelets basada en la transformada de Haar.

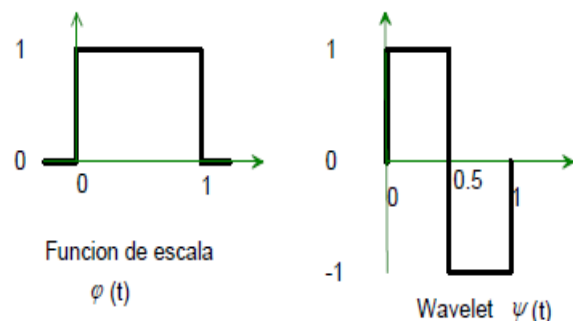


Figura 1: Transformada Wavelet

Haar wavelets asume una señal discreta (o señales analógicas discretizadas, DTW) y toma 2^n muestras equitativas, descompuestas en dos sub-señales con la mitad de longitud, estas señales son promedios, actúa como filtro paso bajo; y diferencias, actúa como filtro paso alto.

$$f = (f_1, f_2, \dots, f_N);$$

$$f_1 = g(t_1), \quad f_2 = g(t_2), \quad \dots, \quad f_N = g(t_N) \quad (1)$$

Para el promediado:

$$c_m = \frac{f_{2m-1} + f_{2m}}{\sqrt{2}}, \quad m = 1, 2, 3, \dots, N/2; \quad (2)$$

En cuanto a las diferencias:

$$d_m = \frac{f_{2m-1} - f_{2m}}{\sqrt{2}}, \quad m = 1, 2, 3, \dots, N/2; \quad (3)$$

Siguiendo con esto se realizan Análisis de Multiresolución (MRA), transformando las imágenes captadas en matrices de sencilla comparación y computo.

$$f = f_1(1, 0, \dots, 0) + f_2(0, 1, 0, \dots, 0) + \dots + f_N(0, 0, \dots, 1). \quad (4)$$

Teniendo una base natural

$$\begin{aligned} V_1^3 &= (1, 0, \dots, 0); \\ V_2^3 &= (0, 1, 0, \dots, 0); \\ &\vdots \\ V_N^3 &= (0, 0, \dots, N); \end{aligned} \quad (5)$$

Por lo tanto

$$f = f_1 V_1^3 + f_2 V_2^3 + \dots + f_N V_N^3; \quad (6)$$

Quedando c y d como vectores de coeficientes.

Para generar el patrón a buscar en la imagen, se utiliza el algoritmo de Viola-Jones [5], con el cual para crear el clasificador emplea Ada-Boost, donde escoge en cada nodo de la red neuronal, a crear mediante aprendizaje y entrenamiento.

Ada-Boost, se tiene como ejemplo,

$$\{x_i, y_i\}_{i=1}^m, x_i \in X, y_i \in \{-1, 1\} \quad (7)$$

aumentando cada ronda en t

$$\sum_i D_t(i) = 1 \rightarrow D_1(i) = \frac{1}{m}; \quad (8)$$

obtiene un aprendizaje débil

$$h_t(x) \in \{-1, 1\} \quad (9)$$

Se tiene el error como

$$e_t = \sum_{i=1}^n D_t(i) [|h_t(x_i) \neq y_i|] \quad (10)$$

Acaba si no cumple $e_t < 0.5$

Las aportaciones como

$$\alpha_t = \frac{1}{2} \ln \frac{1 - e_t}{e_t} \quad (11)$$

Modificando $D_t(i)$; estos procesos se repiten T veces. Para obtener en la salida un aprendizaje fuerte tal que:

$$H(x) = \text{sign} \left(\sum_{t=1}^T \alpha_t h_t(x) \right) \quad (12)$$

Para entrenar esta red se obtienen imágenes negativas, las que no contienen el patrón a distinguir, y positivas, imágenes con el patrón. En las últimas la figura u objeto a reconocer irá cambiando de posición, orientación y demás parámetros configurables, datos conocidos por el algoritmo para así ‘aprender’ a buscar un conjunto parecido.

OpenCV incluye las funciones necesarias para crear estos clasificadores en formato ‘.xml’ siguiendo un proceso sencillo pero lento. Se utilizará una imagen positiva (patrón) para crear más insertándola en las negativas, mediante el comando “createsamples”, nos devuelve las imágenes positivas y la posición del patrón que posteriormente servirá para el entrenamiento mediante el comando “haartraining”, que efectuará los cálculos para proporcionar el clasificador. Ambos comandos poseen opciones de funcionamiento necesarios para una correcta creación del ‘.xml’.

Retomando el uso de la función HaarDetectObjects comentar que de los parámetros utilizados encontramos, image que es la imagen donde detectar el patrón, cascade el clasificador, storage el espacio reservado en memoria, el factor de escala, el mínimo número de patrones vecinos, flags bit que permite el uso de opciones y modos de funcionamiento y minSize marca el tamaño mínimo para el patrón.

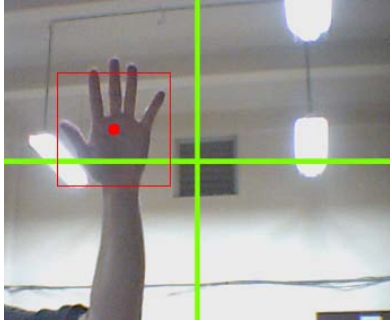


Figura 2: Patrón detectado

Para el proyecto se detectará la mano del usuario, siendo válido para cualquier persona que ocupara el puesto de control (rango de visión de la cámara) y obtendrá las coordenadas del patrón recuadrándolo y calculando su centro, ya que la posición del manipulador es conocida y fija respecto a los ejes creados se procede al cálculo de los movimientos, mediante ecuaciones de cinemática inversa, para posicionar la pinza en el centro dado. Por otro lado tenemos la tercera dimensión la cual podemos medir su variación comparando el tamaño del patrón a medida que aumenta o disminuye si acercamos o alejamos la mano del objetivo. Así se consigue controlar desplazamiento en el eje Z de una manera intuitiva ya que mantiene la similitud entre el gesto y la acción mejorando así la interacción hombre-máquina.

3.2 SIMULACIÓN DEL POSICIONAMIENTO ESPACIAL PARA CONFIGURACIÓN DE ROBOT SCARA

El modelo de posicionamiento desarrollado mediante el sistema de visión artificial se ha simulado para conseguir una representación gráfica de la configuración del robot SCARA. En ésta simulación se ha utilizado OpenGL, software libre que permite dibujar en 2D y 3D, pudiendo así trabajar sin necesidad de conectar el SCARA; esta simulación del espacio de trabajo y su respuesta ha permitido poder evitar conflictos a la hora de su implementación física, habituarse al manejo del sistema resulta más fácil si se dispone de este simulador.

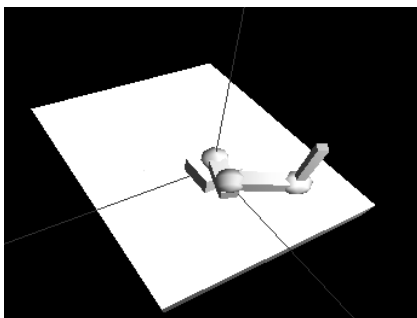


Figura 3: Simulación 3D del robot SCARA

Principalmente se ha utilizado la simulación a la hora de depurar código evitando también el desgaste innecesario de los componentes del robot.

4. APLICACIÓN DEL INTERFAZ EN UN ROBOT SCARA

4.1 MORFOLOGÍA DEL ROBOT SCARA

La arquitectura SCARA es la empleada para la configuración del manipulador, ya que conforma una estructura ergonómica y su control es relativamente simple, combinando así, su versatilidad física y su programación con un moderado grado de complejidad debido a que no poseen demasiados grados de libertad.

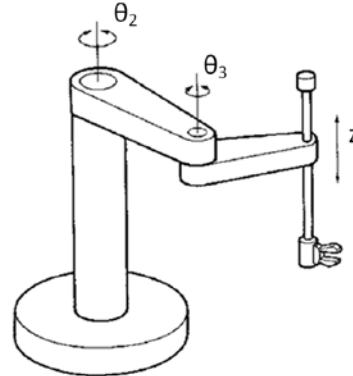
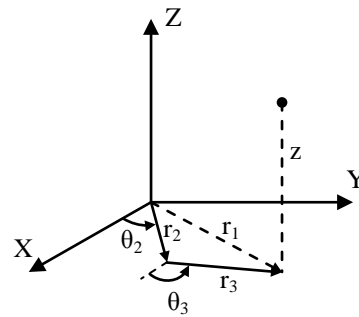


Figura 4: Configuración SCARA

Esta arquitectura está dotada de tres grados de libertad y se caracteriza por la total libertad de movimiento alrededor de los 360° de los ejes X e Y, en nuestro caso limitada por las características mecánicas de los servomotores.

El robot SCARA desarrollado está compuesto por tres eslabones unidos por medio de dos articulaciones rotacionales y una prismática. Las rotacionales cuentan con un giro de 180° generado por el servomotor y una prismática responsable de la componente Z que desliza la pinza hasta alcanzar su objetivo.

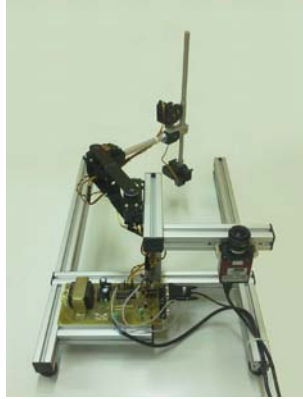


Figura 5: Robot SCARA

Los servomotores de las articulaciones rotacionales están localizados de forma que se obtiene el mayor espacio de trabajo posible y que la accesibilidad a los puntos es la adecuada.

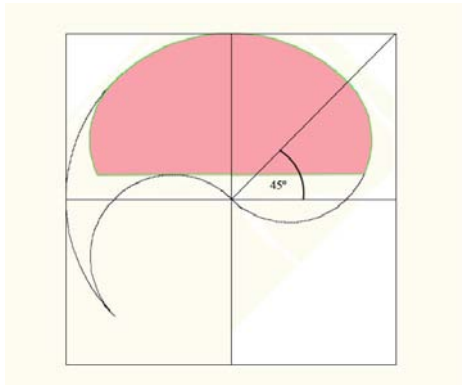


Figura 6: Desplazamiento de 45° para la optimización del espacio de trabajo

4.2 CONTROL DEL ROBOT SCARA

El diseño hardware se ha implementado a través de un microcontrolador con el cual generamos la señal PWM necesaria para el movimiento y posicionamiento de los servomotores.

Esta señal esta generada a través de los temporizadores del microcontrolador, multiplexando esta señal llegamos a obtener hasta 8 señales PWM independientes y servibles para el control de los servomotores.

Estas señales, debido a las características mecánicas de los servomotores elegidos (HS-645MG y HS-422 de HITEC), seguirán un protocolo marcado por el fabricante, son de sencilla implementación en cualquier lenguaje de programación o generador de pulsos.

El ancho de pulso mínimo generado para que el servomotor se posicione en uno de sus extremos debe ser de 0.3ms el cual corresponderá a 0°, siendo su opuesto (180°) el equivalente a un ancho de pulso de 2.1ms (fig. 7). Esta señal debe ser refrescada con un máximo de 20ms para un correcto posicionamiento y anclaje del servomotor en la posición deseada.

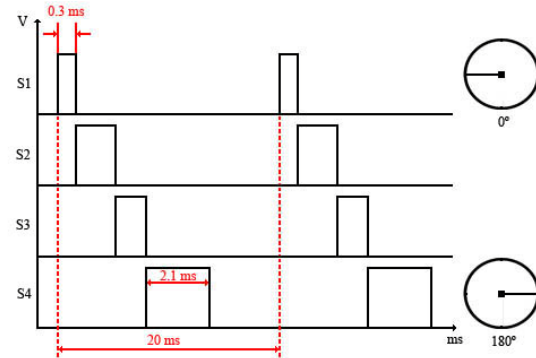


Figura 7: Señal PWM multiplexada para el posicionamiento de los servomotores

De los modelos geométricos más comúnmente utilizados, se opta por resolver el Modelo Geométrico Inverso del robot SCARA mediante las ecuaciones que a continuación se detallan.

$$r1 = \sqrt{r1x^2 + r1y^2} \quad (13)$$

$$\theta1 = \begin{cases} \text{atan}\left(\frac{r1y}{r1x}\right) & \text{if } r1 > 0 \\ 180^\circ + \text{atan}\left(\frac{r1y}{r1x}\right) & \end{cases} \quad (14)$$

$$\theta2 = \theta1 - \text{acos}\left(\frac{r1^2 + r2^2 - r3^2}{2 \cdot r1 \cdot r2}\right) \quad (15)$$

$$\theta3 = \theta1 - \text{acos}\left(\frac{r2^2 - r1^2 - r3^2}{2 \cdot r1 \cdot r3}\right) \quad (16)$$

La solución de las ecuaciones (13), (14), (15) y (16), han sido implementadas en lenguaje C para su resolución, debido a su fácil uso y velocidad de cómputo, el resultado de estas forman parte de una trama datos que consta de dos Bytes. Esta trama indica el servomotor a controlar y la posición a situarse. Los datos son enviados vía comunicación serie a la placa de control, se opta por este tipo de comunicación por ser de fácil configuración y rápida transmisión.

4 CONCLUSIONES Y LINEAS FUTURAS DE TRABAJO

Mediante esta técnica, se obtienen resultados muy favorables de cara al desarrollo de este tipo de sistemas o ambientes. Primeramente, por la sencillez de diseñar un sistema de estas características, la disponibilidad de material y el fácil acceso a tecnologías suficientemente avanzadas como para formar parte de este tipo de aplicaciones. La facilidad de control y la sencillez de los movimientos que debe realizar el usuario para un manejo completo de este. Debido a que se suprimen los periféricos habituales esto supone un ahorro económico y energético a la empresa que implementa el sistema.

Podemos señalar que gracias a la sinergia entre la visión por computador y los ambientes virtuales (inmersivos o no), junto a las técnicas de realidad aumentada u holográficas (objeto de futuras líneas de trabajo), pasamos de trabajar en ambientes hostiles a entornos con mejores condiciones, más limpios, seguros y cómodos para los operarios. Pudiendo alcanzar nuevas metodologías, técnicas o procedimientos cooperativos entre hombres y robots. Argumentando con esto el concepto de cibermetización virtual de usuarios u operarios.

Como se ha mencionado en los párrafos anteriores, las líneas futuras de trabajo van a estar dirigidas hacia una mejora en el interface de modo que la interacción sea de un modo aun más natural si cabe, utilizando para ello técnicas de realidad aumentada, dispositivos holográficos, etc. Ya que estos posibilitan crear un ambiente virtual con propiedades similares a un ambiente real.

Referencias

- [1] Y. Boada Acosta, L. Morales Escobar, N. G. Sotomayor, “Emsamblaje y control de un robot de tres grados de libertad tipo SCARA(Selective Compliance Assembly Robot Arm)”, Escuela Politécnica Nacional de Ecuador.
- [2] G. Bradski y A. Kaehler (2008) “Learning OpenCV”, O’Reilly Media.
- [3] D. Cervantes Cabrera, J. Larios Delgado, R. Leriche Vázquez y M. C. Ramos Nava, “Creación de ambientes virtuales inmersivos con software libre”, Universidad Nacional Autónoma de México.
- [4] G. A. Galvis, D. Jabba y P. M. Wightman “Ambientes de realidad virtual no inmersiva multiusuario con herramientas de software libre”, Universidad del Norte Barranquilla, Colombia.
- [5] M. J. Jones y P. Viola (2001), “Rapid Object Detection Using a Boosted Cascade of Simple Features”, IEEE CVPR.
- [6] R. Lienhart y J. Maydt (2002), “An Extended Set of Haar-like Features for Rapid Object Detection”, IEEE ICIP, pp. 900-903.
- [7] A. Ollero (2001), “Robótica. Manipuladores y robots móviles”, Marcombo Boixareu, ISBN: 8426713130.