

BRAS: una herramienta para la experimentación con una celda robótica realimentada por visión

Toni Burguera, Gabriel Oliver, Alberto Ortiz

Dept. de Matemàtiques i Informàtica
Universitat de les Illes Balears (UIB)
e-mail: {goliver,dmiaor0}@clust.uib.es

Resumen

Debido a la relevancia de las celdas robóticas en el mundo industrial actual, en el Departamento de Matemáticas e Informática de la UIB se ha desarrollado BRAS, una herramienta *software* que permite a los alumnos de las asignaturas de robótica de la Ingeniería Superior de Informática y de la Ingeniería Técnica Industrial de la UIB experimentar directamente con una celda robótica genérica. Esta celda consta de un brazo robótico, una cinta transportadora, una cámara de video para uso industrial, una tarjeta de adquisición de imágenes IV-400 de muTech y un ordenador compatible PC. BRAS está actualmente configurada para los componentes concretos del entorno descrito, aunque su adaptación a cualquier otro es sumamente sencilla debido a la arquitectura con que ha sido diseñada y el uso de programación orientada a objetos.

1. Introducción

Las cadenas productivas industriales incluyen, en la mayoría de los casos, puntos que resultan inadecuados para el trabajo humano, o bien tareas altamente repetitivas que no necesitan de una supervisión muy compleja. En muchos de estos casos el proceso a ejecutar consiste en clasificar objetos, detectar y/o separar objetos defectuosos, o ensamblar piezas. Todos estos aspectos convierten a las celdas robóticas en herramientas útiles para mejorar la producción de la planta y la seguridad de los trabajadores, razón por la cual están ampliamente difundidas en los entornos industriales actuales.

Debido a que cada proceso industrial posee características peculiares, las configuraciones

particulares de las diferentes celdas robóticas pueden variar. Sin embargo, siempre subyace la idea de un manipulador controlado por programa que realiza tareas sobre objetos que circulan sobre una cinta transportadora. Además, si el proceso requiere la identificación de dichos objetos, o partes de ellos, o de algunas de sus propiedades, como su posición u orientación en la cinta, entonces la celda robótica incorpora también un sistema de procesamiento de imágenes. Por ello, los subsistemas mencionados configuran lo que podríamos definir como una celda robótica genérica.

La necesidad de proporcionar a los alumnos de las asignaturas de robótica de la Ingeniería Superior de Informática y de la Ingeniería Técnica Industrial una formación práctica en robótica industrial ha dado lugar a que los autores desarrollen la herramienta *software* que se presenta en esta comunicación: BRAS, una herramienta para la experimentación con una celda robótica genérica como la mencionada.

El objetivo básico de BRAS es permitir la interacción entre los elementos de la celda robótica. Esto es, es capaz de adquirir imágenes provenientes de una cámara industrial, de analizarlas y de controlar el robot para que, a partir del resultado del análisis visual, se lleve a cabo la tarea especificada. Es trabajo del usuario la especificación de los objetos a identificar, así como de la acción a realizar con cada uno de ellos. Para ello, se dispone de un lenguaje de definición de objetos basado en las propiedades geométricas de los mismos, DOB, y de un lenguaje de especificación de acciones a ejecutar por el manipulador, BSC. Ambos lenguajes son propietarios de esta plataforma de experimentación.

La incorporación de intérpretes de estos dos lenguajes en BRAS permite a los estudiantes

experimentar con diferentes estrategias de especificación de objetos y acciones.

El resto de la comunicación se articula de la siguiente forma: en la sección 2 se presenta la estructura tanto *hardware* como *software* del sistema; en la sección 3 la atención se concentra en la descripción de la configuración y uso del subsistema de visión; la sección 4 está dedicada a la descripción y configuración del subsistema de actuación; la sección 5 hace referencia a la calibración de la celda robótica; y, finalmente, la sección 6 presenta algunas conclusiones. En esta comunicación no se pretende hacer una descripción exhaustiva de BRAS. Para ello, se recomienda consultar la guía del usuario que acompaña a la aplicación.

2. Entorno de trabajo y estructura de BRAS

En la figura 1 se presenta una descripción gráfica de la celda robótica con la que interactúa BRAS.

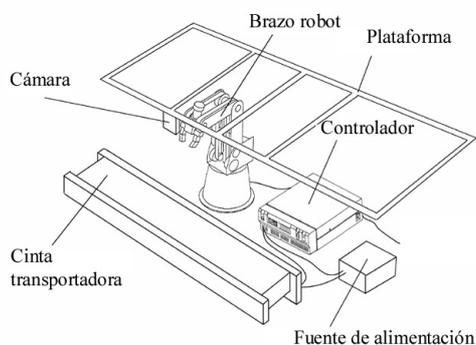


Figura 1. Descripción de la celda robótica

La configuración particular de la celda consiste en los siguientes elementos: (a) un brazo robot Scorbot-ER V Plus [3], dotado de 5 grados de libertad más una pinza, y de un controlador multitarea programable en ACL [2] capaz de gestionar 8 grados de libertad, de los cuales ya tiene cubiertos 6 con las cinco articulaciones del brazo y la pinza; (b) una cinta transportadora, que puede ser configurada en uno de dos estados, activa o inactiva, a través de una de las salidas libres del controlador del brazo; (c) un ordenador PC con Windows NT, conectado mediante RS-

232 con el controlador del brazo, para transmitir órdenes tanto al brazo como a la cinta; y (d) un sistema de adquisición de imágenes, consistente en una cámara B/N con salida PAL, suspendida de una plataforma metálica que se encuentra sobre el espacio de trabajo del brazo, y una tarjeta de digitalización IV-400 de muTech [1] instalada en el PC y capaz de capturar señal PAL y NTSC.

En cuanto a la estructura de BRAS, ésta consiste básicamente de dos módulos: uno de análisis visual de la escena, y otro que controla la actuación del brazo de acuerdo con lo expresado por el usuario en un archivo BSC. Además hay un módulo de calibración del sistema, a través del cual se ajustan los parámetros de operación y se realizan las transformaciones correspondientes de las descripciones proporcionadas por el módulo de visión a descripciones comprensibles por el módulo de actuación.

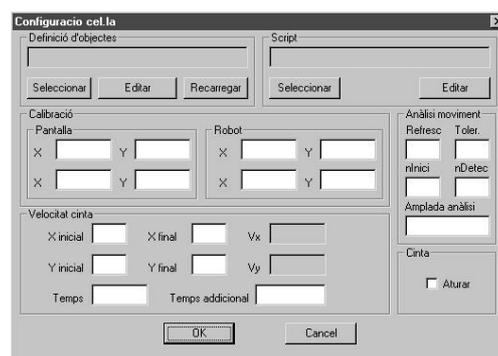


Figura 2. Ventana de configuración de la celda robótica

Todo el sistema ha sido programado en C++ utilizando orientación a objetos y con el objetivo de conseguir un sistema altamente modular y fácilmente mantenible. Para facilitar la adaptación de la aplicación, los nombres de las diferentes clases siguen una serie de normas. Todas comienzan con la letra C mayúscula seguida de prefijos de tres o seis letras que indican la funcionalidad y el nombre concreto de la clase. En particular, las clases con prefijo *Dsp* corresponden a objetos que interactúan directamente con un dispositivo, como la cámara o el robot. Esto implica que para reconfigurar la celda robótica con otros dispositivos físicos basta con realizar las modificaciones y añadidos oportunos a las clases *Dsp*.

La figura 2 muestra la ventana de configuración de la aplicación, a través de la que, en particular, se especifican los ficheros de definición de objetos y de actuación para la tarea a ejecutar. La aplicación está enlazada con el editor externo de dominio público PFE a través del que se puede editar ficheros DOB y BSC.

3. El subsistema de visión

El análisis visual de la escena comienza con la eliminación de ruido e información no útil de la imagen, para pasar a continuación a un análisis de conectividad para determinar el número de objetos de la escena, y finalmente a la extracción de características de dichos objetos para su posterior identificación. La entrada de este subsistema es por tanto la imagen capturada por la cámara y la salida es la lista de objetos presentes en la escena. Para ello, se requiere que los objetos aparezcan aislados en la imagen original.

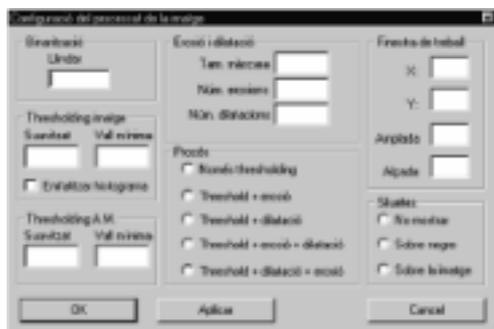


Figura 3. Configuración del subsistema de visión

El proceso de eliminación de información no útil transforma la imagen original en una imagen más sencilla en la que se ha reducido el número de tonalidades de gris al número de colores distintivos de la escena. Esto se puede conseguir mediante *binarización*, cuando sólo hay un objeto en la escena; o bien mediante *umbralizado*, cuando hay más de un objeto en la escena.

En cuanto a la eliminación de ruido, se hace uso de las técnicas de procesado morfológico *erosión* y *dilatación*. De esta manera, se puede eliminar agrupaciones aisladas de píxeles que son atribuibles al ruido presente en la imagen debido a

especularidades, diferencias en la reflectancia de los objetos de la escena, defectos de la cámara, etc.

Una vez se tiene una imagen simplificada y sin ruido, se procede al análisis de conectividad para efectivamente determinar las regiones aisladas presentes en la imagen y así pasar a identificar objetos concretos.

La figura 3 muestra el cuadro de diálogo a través del cual se especifica los parámetros de los procesos descritos.

El subsistema de visión también lleva a cabo el análisis de movimiento en el caso de que la cinta transportadora esté activa. En este caso se trata de detectar la entrada de objetos en la escena, para lo cual se analiza un cierto número de columnas situadas a la derecha de la imagen. Los parámetros de configuración aparecen en el cuadro de diálogo de la figura 2, en el panel denominado *anàlisi de moviment*.

Compacitat	Area	Per	(xc,yc)	(Dmin,Dmax)	(xm,ym)	(xM,yM)	Orien	
1.500000	12745	496	(171,171)	38	163	286,163	277,163	0.816164
1.000000	9198	487	(249,163)	29	130	(273,163)	(273,287)	2.987493
1.070000	26476	1071	(416,284)	55	126	(362,284)	(387,136)	1.464327
0.750000	9023	262	(158,257)	42	46	(122,226)	(152,369)	0.926217

Figura 4. Visualización de las características de los objetos detectados

El proceso de extracción de características obtiene descripciones de los objetos de la escena útiles para su identificación y muestra los valores detectados para cada objeto de la escena, en coordenadas imagen, en una ventana como la mostrada en la figura 4, donde: *Area* es el número de píxeles que forman parte del objeto, *Per* es el número de píxeles que constituyen el perímetro del objeto, (x_c, y_c) son las coordenadas del centro de gravedad, (D_{min}, D_{max}) son las distancias *Manhattan* mínima y máxima entre el centro de gravedad y los píxeles del perímetro, (x_m, y_m) son las coordenadas del píxel del perímetro más cercano al centro de gravedad, (x_M, y_M) son las coordenadas del píxel del perímetro más alejado del centro de gravedad, *Orien* es la arcotangente del eje de mínima inercia del objeto, y *Compacitat* es la relación entre el perímetro al cuadrado y $4\pi Area$.

La conexión entre el subsistema de visión y el subsistema de actuación se realiza a través de un fichero de definición de modelos de objetos DOB, en el cual se listan y describen, mediante lenguaje DOB, los modelos de los objetos relevantes para la tarea en ejecución. Las descripciones consisten en grupos de condiciones sobre las características que el sistema calcula sobre cada objeto. Todas las condiciones han de ser satisfechas simultáneamente para que el objeto observado sea considerado compatible con el modelo y el subsistema de actuación ejecute acciones sobre él.

4. El subsistema de actuación

Una vez el subsistema de visión ha analizado la escena, la información obtenida es enviada al subsistema de actuación para que se tomen las decisiones sobre qué ha de hacer el robot. Las órdenes se envían a través de una conexión serie con el controlador del robot empleando lenguaje ACL.

El subsistema de actuación puede operar en uno de tres modos en relación con la cinta transportadora: *con la cinta parada, con la cinta en movimiento y parada cuando se detecta un objeto, y con la cinta continuamente en movimiento.*

El diálogo directo con el controlador del brazo es posible desde el subsistema de actuación, lo cual puede resultar útil para activar, por ejemplo, la cinta transportadora. Sin embargo, lo ideal es poder ejecutar una tarea concreta de forma automática. Para ello, se dispone de un lenguaje tipo *script* denominado BSC que permite especificar conjuntos de acciones a realizar por el brazo para cada uno de los objetos detectados por el subsistema de visión.

El flujo de ejecución de un *script* BSC está estrechamente ligado a los objetos que ha identificado el módulo de visión. Esto es, el *script* se puede ejecutar tantas veces como objetos han sido detectados en la escena. En cada ocasión, el flujo de ejecución es dirigido hacia las acciones correspondientes al objeto en cuestión a través de sentencias condicionales.

5. Calibración del sistema

La interacción entre los componentes de la celda robótica requiere el ajuste de algunos parámetros. Por un lado, la información proporcionada por el subsistema de visión no es directamente utilizable por el subsistema de actuación debido a que trabajan en sistemas de coordenadas diferentes. La matriz de transformación de un sistema en otro se puede calcular conociendo las coordenadas de dos puntos en los dos sistemas de coordenadas, que o bien el usuario introduce manualmente o bien se correlacionan de forma interactiva.

Por otro lado, cuando se trabaja con la cinta transportadora continuamente en movimiento es necesario saber su velocidad para poder anticipar la ubicación de los objetos tras los tiempos de análisis de la imagen y de desplazamiento del brazo. La calibración se realiza de forma interactiva determinando el tiempo que transcurre entre que un objeto entra en el campo de visión y éste sale del mismo.

6. Conclusiones

Se ha presentado una herramienta para la experimentación con una celda robótica genérica llamada BRAS. La herramienta es altamente parametrizable para poder adaptarse a multitud de entornos. En la misma línea, en caso de necesitar interactuar con otros dispositivos físicos, sólo es necesario modificar la E/S de bajo nivel.

En cuanto a su versatilidad, la versión de BRAS que acompaña esta comunicación incorpora diversos experimentos de clasificación de piezas, montaje e inspección, en cada uno de los tres modos de funcionamiento (cinta parada, cinta en movimiento y parada, y cinta en movimiento).

Referencias

- [1] Mutech Corporation. *Image/VGA-400 User's Guide*, 1997
- [2] Orpi S.L. *ACL, Lenguaje de Control Avanzado, v1.43, F.44. Guía de referencia para Controlador-A*
- [3] Orpi S.L. *Scorbot-ER V Plus. Manual del Usuario*