

PROTOTIPO DE VISIÓN ARTIFICIAL DE BAJO COSTE PARA DESARROLLO DE PRÁCTICAS DOCENTES

José Félix González Rojo

Universitat Politècnica de València, jugonro3@etsid.upv.es

Joan Carles Puchalt Rodríguez

Universitat Politècnica de València, juapucro@upv.es

Eugenio Ivorra Martínez

Universitat Politècnica de València, euivmar@upvnet.upv.es

Antonio José Sánchez Salmerón

Universitat Politècnica de València, asanchez@isa.upv.es

Resumen

En el presente artículo se desarrolla una propuesta de prototipo de bajo coste que busca aumentar la motivación y rendimiento de los estudiantes de grado o postgrado mediante la aplicación práctica, en el mismo, de los conceptos adquiridos durante su formación. Su diseño de bajo coste y filosofía abierta permiten la completa modulación del prototipo, facilitando el mantenimiento y la adaptación a cualquier necesidad académica de las disciplinas de la rama de la ingeniería relacionada con la robótica, la electrónica, la automatización, la visión artificial y la informática industrial. La propuesta consiste en un robot cartesiano de tres ejes inspirado en las impresoras 3D actuales que implementa un control en bucle cerrado mediante visión artificial para el seguimiento y la interacción con objetos pequeños.

Palabras Clave: Bajo coste, visión por computador, formación, modular, open software, open hardware, formación postgrado, bucle cerrado, impresión 3D.

1 INTRODUCCIÓN

La formación académica en las carreras universitarias técnicas viene complementada con sesiones prácticas para la aplicación de los conceptos presentados en las clases magistrales. Estas sesiones no solo completan la formación, si no que suponen una nueva perspectiva y acercan al alumno aspectos de la disciplina que de otra forma no se adquirirían.

Una de las principales razones desde el punto de vista docente por la que las prácticas son recomendables es

la motivación que despierta en los alumnos. Esto se debe al interés que les genera el poder aplicar el conocimiento teórico adquirido para realizar tareas y solucionar problemas reales como por ejemplo: desarrollar aplicaciones, diseñar sistemas automáticos, programar y manipular robots... Además, la motivación del alumno está directamente relacionada con un mejor rendimiento académico [1].

Sin embargo, muchas veces esta motivación se ve reducida por la falta de medios o por el carácter simulado de la sesión práctica. Un caso particular muy común es, en las asignaturas de visión por computador, el empleo de simuladores o laboratorios virtuales [7,9,10] debido a su flexibilidad y a su bajo coste comparado con los sistemas de visión profesionales. Sin embargo, el uso de simuladores en muchas ocasiones desemboca en el desinterés del alumno por la materia al no ver la utilidad real.

Si bien estos simuladores permiten la abstracción de la, normalmente, compleja y problemática base de instalación, configuración y programación de los sistemas físicos; también carecen de una aplicación física que implementar en un sistema con el que se pueda interactuar y que permita, además, la consolidación de conocimientos de otras ramas íntimamente relacionadas como son la informática, la electrónica o la robótica industrial. De hecho, es en estas ramas donde el uso de una plataforma experimental es fundamental [8].

Normalmente, la adquisición de un equipo o material de laboratorio conlleva un gasto elevado, a veces prohibitivo, para la institución, siendo este equipo, normalmente, cerrado y únicamente compatible con su propio software/hardware, impidiendo la comunicación con otros dispositivos o la programación mediante otros lenguajes. No obstante,

en los últimos años se ha desarrollado enormemente una filosofía que ha ido despertando el interés de más y más personas por la tecnología y la ingeniería como es el movimiento Maker basado en el DIY (*Do It Yourself*). Hoy en día el software y el hardware libre son muy populares, facilitando el acceso a soluciones abiertas y completas, reduciendo el coste de implementación de las mismas notablemente y permitiendo la modificación del mismo a voluntad. Éstas son características con grandes beneficios desde el punto de vista académico.

Considerando estos factores, el presente proyecto introduce, apoyándose en esta nueva filosofía abierta, el desarrollo de un prototipo multidisciplinar de bajo coste diseñado e implementado en hardware y software libre, fácilmente adaptable para la realización de prácticas de laboratorio en las disciplinas de visión por computador y robótica.

El prototipo implementado realiza el control en bucle cerrado de una impresora 3D mediante visión por computador, e ilustra mediante un proyecto multidisciplinar basado por entero en la visión por computador las posibilidades de implementación práctica a bajo coste de este sistema.

Tras esta introducción, el artículo se organiza como sigue. En el punto 2 se describen los elementos que componen el prototipo. En el punto 3 se presentan aplicaciones prácticas del mismo en forma de ensayos que hacen uso de conceptos de control automático y visión por computador, como propuestas de uso. Por último, en los puntos 4 y 5, se muestran los resultados de los ensayos, acompañados por conclusiones respecto al prototipo.

2 SISTEMA MATERIAL E IMPLEMENTACIÓN

El prototipo debe ofrecer las características indicadas en la introducción al menor coste posible. Tras el análisis de distintas posibilidades se detallan en los siguientes subapartados las soluciones que se han adoptado para los distintos componentes del sistema: hardware (estructura y actuadores, microcontrolador y cámara e iluminación) y software (lenguajes de programación, librerías para visión por computador y librerías de interfaz GPIO y comunicación).

2.1 HARDWARE

2.1.1 Estructura y actuadores

Existen diferentes posibilidades para construir una plataforma de prácticas, pero en este proyecto, el sistema implementado se basa en el robot cartesiano de tres ejes lineales cuya implementación se ha

popularizado de la mano de la impresión 3D, ya que es la base detrás de la inmensa mayoría de modelos de estos productos [4].

El principal motivo para elegir este sistema de entre el amplio abanico de posibilidades es el reducido coste de construcción y el carácter abierto de la comunidad que lo ha creado y popularizado durante los últimos años. Además, una impresora 3D constituye una solución completa desde el punto de vista de la robótica, incluyendo estructura, actuadores, transmisión y microcontrolador con etapa de potencia.

En cuanto a estructura se refiere tanto tamaño como elementos que la componen se pueden escalar y personalizar de acuerdo a las necesidades docentes. Cualquier elemento puede incluirse con una pieza impresa diseñada para tal efecto de manera sencilla.

En cuanto a los actuadores, la configuración usual de las impresoras 3D utiliza motores paso a paso NEMA17 que permiten una implementación eficaz sin la necesidad de *encoder* u otro sensor que permita el control en bucle cerrado necesario para posicionar los ejes.

La transmisión de movimiento a los ejes se realiza mediante poleas y correas dentadas del estándar GT2. La figura 1 ilustra esta disposición.



Figura 1: Motor paso a paso NEMA 17. Y polea con correa GT2

El control de cada motor se realiza a través de *drivers* A4988, normalmente conocidos como Pololus. Estos *drivers* se sirven de una etapa de potencia conocida como RAMPS en su versión 1.4. El *firmware*/proyecto libre Marlin, un controlador completo de varias configuraciones distintas de máquinas de control numérico por computador que puede modificarse y adaptarse a cualquier tipo de disposición de la estructura y los actuadores, utiliza un sistema de comandos para aplicar cualquier cambio en la máquina, desde consignas y parámetros de aceleración y velocidad, hasta el desplazamiento lineal en un eje.

El código de movimiento lineal en un eje es: G0 E d n

Donde E es el eje a mover (X, Y o Z) y d (+d o -d) es la distancia que queremos movernos. Por ejemplo: G0 Y12n, Movería el carro Y 12mm en el sentido positivo del eje.

Hay que destacar la importancia del final de carro ("n") ya que es el elemento que interpreta Marlin como final de instrucción y sin el cual ningún comando llegaría a ejecutarse.

Un listado de los comandos con su descripción puede consultarse en [13].

2.1.1 Microcontrolador

El microcontrolador que gobierna la etapa de potencia e implementa el firmware Marlin antes mencionado es un Arduino Mega con chip ATmega2560 (Figura 2 izq.).



Figura 2: Arduino Mega con RAMPS V.1.4. y Raspberry Pi 2

Si bien toda la estructura y su control podrían aprovecharse tal y como se acaban de presentar, el verdadero cerebro de nuestro prototipo será el SBC (Simple Board Computer) Raspberry Pi, concretamente la versión Raspberry Pi 2 (Figura 2 der.). Este ordenador permite utilizar distintos sistemas operativos previamente compilados para funcionar con la arquitectura de su procesador, entre ellos, nosotros utilizaremos Raspbian, un sistema en base Linux concebido especialmente para este dispositivo. Una de las mayores ventajas de la Raspberry Pi, además de constituir un ordenador completo al precio de cualquier microcontrolador tipo Arduino, es la enorme comunidad que la respalda creando y compartiendo infinidad de proyectos y librerías que suponen una fuente cada vez más grande de referencias de la que beber a la hora de embarcarse en cualquier proyecto. Además, un sistema de base Linux como Raspbian supone completa transparencia para el usuario de todo proceso que se decida implementar. Todo esto nos permite abstraernos de la capa de implementación y programación del hardware, pudiendo enfocarse directamente en su control y aplicación.

2.1.1 Cámara e iluminación

Para la implementación de proyectos de visión artificial es necesaria una cámara que permita la

adquisición. Normalmente este elemento alcanza grandes dimensiones lo que dificulta su acople en los montajes de laboratorio. Se utilizará una cámara de la misma fundación Raspberry Pi, la Raspberry Pi Camera Module de 5Mpix. y una resolución máxima de 2592x1944 (Figura 3), cuya lente se puede regular con ayuda de unas pinzas para ajustar el enfoque. Éste supone un pequeño elemento fácilmente adaptable a cualquier pieza. Además de estar concebida para su uso con la Raspberry Pi, el bus de cable flexible que utiliza para comunicarse puede adquirirse con distintas longitudes, facilitando el movimiento de la cámara en la impresora y reduciendo el cableado del prototipo. Además la conexión y desconexión del mismo es sencilla y segura.

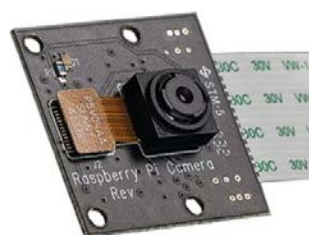


Figura 3: Raspberry Pi Camera NOIR

Para implementar un sistema de visión artificial completo no basta sólo con la cámara. Un elemento tan importante como ella es el control de la iluminación que puede facilitar y mejorar enormemente la adquisición, reduciendo el preprocesamiento, y con ello, la dificultad del proyecto al mínimo. Existen distintas soluciones y configuraciones para llevar a cabo esta solución, desde complicados sistemas de lentes y espejos, a distribuciones de LEDs en torno a la cámara.

Como ya se ha comentado, una de las principales ventajas de este prototipo es la modularidad, por lo que se ha optado por una solución diferente a las usualmente utilizadas y que puede fácilmente desinstalarse o utilizar con un propósito distinto. Esta ventaja consiste en utilizar una pantalla como base de trabajo, de esta manera podremos variar el tipo de iluminación. Para este proyecto se utilizará entre el abanico de posibilidades, de nuevo, una de la casa Raspberry Pi, el Raspberry Pi *touchscreen display* de 7 pulgadas y resolución de 800x480.

Como elemento adicional, se decide disponer de una luz frontal mediante una linterna LED o un puntero láser que permita aún mayor variedad en técnicas de iluminación. Para ello recurriendo, una vez más, a la facilidad de modificación del prototipo, se ha diseñado una pieza que soporte la linterna, que le permita acoplar la cámara y que disponga de un elemento para interactuar con objetos. La figura 4 muestra un detalle de esta pieza.

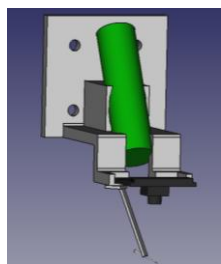


Figura 4: Pieza soporte de la herramienta.

La figura 5 ilustra el diseño y montaje del sistema de adquisición con la pieza anteriormente mostrada y la pantalla con el filtro de cristal esmerilado.

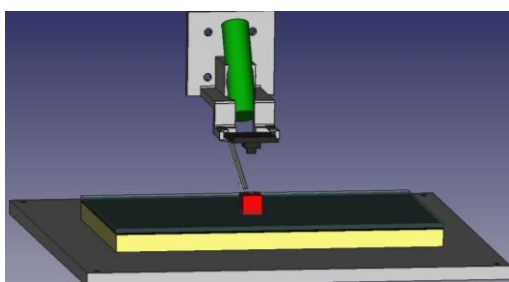


Figura 5: Sistema de adquisición

2.2 SOFTWARE

2.1.1 Lenguaje de programación

El lenguaje de programación que se emplee en los proyectos dependerá enormemente de la función de los mismos y del usuario que los desarrollará.

En este caso se ha optado por C++ como lenguaje de programación orientado a objetos con capacidad para desarrollar cualquier tipo de aplicaciones y de gran compatibilidad con C, siendo el conjunto de ambos el lenguaje en el que se desarrollan la mayoría de recursos de carácter libre, además de disponer de enormes similitudes con otros lenguajes como el que utiliza Arduino.

El lenguaje de programación a utilizar nos determina el compilador. En nuestro caso se utilizará gcc del proyecto GNU, una vez más por su filosofía de software libre. Llegados a este punto se puede optar por una programación directa y arriesgada basada en programación mediante un editor de texto y compilación manual a través de algún recurso como CMake. O puede utilizarse un IDE de cualquier naturaleza de los disponibles para el sistema operativo que utilice.

2.1.1 Visión Artificial

En cuando a proyectos de visión por computador se refiere, existen multitud de librerías en las cuales

apoyarse. En nuestro caso la filosofía de código abierto y libre, junto a la comunidad que la apoya, ha hecho de OpenCV [5] nuestra mejor opción.

OpenCV es compatible tanto con Python como con C/C++.

2.1.1 GPIO interface and communication

Una de las principales necesidades para la puesta en funcionamiento del prototipo es establecer una comunicación entre Raspberry y Arduino para poder controlar el robot desde nuestro sistema operativo mediante el envío de código G.

Para ello, se ha utilizado la librería de código abierto Wiring Pi de Gordon Henderson [6], que permite configurar, inicializar y utilizar cualquier tipo de conexión que permita la placa. Para el prototipo se ha optado por una implementación sencilla mediante comunicación serie a través del puerto USB de ambos dispositivos.

3 PROPUESTAS DE ENSAYOS

Existen una gran cantidad de aplicaciones sobre las que se puede preparar prácticas con este prototipo. Por ejemplo aplicaciones básicas de procesamiento y segmentación de imágenes [2 y 3], seguimiento de objetos, reconocimiento de objetos, calibración de cámaras [14] y visión 3D [15] o incluso aplicaciones más complejas como por ejemplo fotogrametría [11, 12], micro-inspección y micro-manipulación [16, 17, 18 y 19]. Para comprobar la eficacia del prototipo se proponen a continuación una serie de ensayos que podrían ser prácticas de asignaturas de robótica y visión artificial.

3.1 SEGUIMIENTO DE OBJETOS

Se propone el control y el posicionamiento de la herramienta del prototipo (en este caso la cámara) de manera que un objeto de interés se encuentre siempre centrado en la imagen.

Supongamos el caso más genérico: queremos seguir el objeto más cercano al centro de la imagen.

Para resolver el problema los estudiantes dispondrán de un actuador: la impresora; un sensor: la cámara; y una acción de control: el código G.

Sirviéndose de OpenCV se puede implementar un algoritmo que adquiera imágenes, identifique el objeto de estudio, obtenga su posición, y calcule la acción de control a realizar mediante la diferencia de la posición del objeto y el centro, utilizando la

relación de tamaño del pixel de la cámara con la distancia real, como control proporcional.

Algoritmos de reconocimiento de objetos como puede ser el *blob detector* o el *find contours* que incluyen las librerías de OpenCV supondrán una solución directa a la detección y obtención de la posición.

El cálculo del tamaño de un pixel en distancia real puede obtenerse, por ejemplo, obteniendo una imagen de papel milimetrado.

Un aspecto importante a tener en cuenta es el tiempo de ejecución. Si tomamos imágenes a una resolución elevada tendremos mucha más información del proceso y por tanto mayor precisión, pero esto implica el análisis de un mayor número de píxeles, lo que aumenta drásticamente el tiempo de procesado.

Una forma de mantener la precisión que aporta la máxima resolución sin sacrificar tiempo de proceso, se podría definir una región de interés de manera que todo el proceso que hemos comentado se realice solo en esa región en lugar de en toda la imagen.

Hemos de tener en cuenta que la precisión de movimiento de los ejes viene limitada por los pasos que los motores puedan dar. Ya que la precisión de un píxel es mayor que la precisión de los motores tendremos un error de posición fijo.

3.2 INTERACCIÓN CON OBJETOS

Si bien el seguimiento es una implementación completa que abarca aspectos de visión y control muy interesantes, una posibilidad también atractiva es la de interactuar con el objeto de interés.

Si trabajamos con las resoluciones más altas podremos observar y trabajar con objetos muy pequeños. Los ensayos implementados en este proyecto tienen como objeto de interés una resistencia SMD 0402 (Surface Mount Device) de apenas unas décimas de milímetro de tamaño, como puede verse en la Figura 6. La segunda propuesta es tocarla con la punta de la cerda de un pincel que hará de actuador.

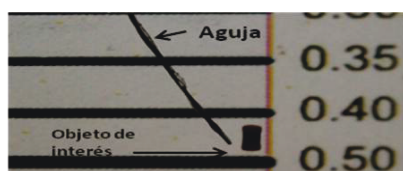


Figura 6: Tamaño del objeto de ensayo donde el número indica el grosor de la línea en mm.

Tocar el objeto es tan sencillo como, una vez centrado el objeto con el proceso de seguimiento de

objetos, bajar una determinada altura en el eje Z y después subirla. Si queremos empujar el objeto, primero nos moveremos un número de pasos hacia la dirección contraria en que lo queremos empujar, bajaremos la misma altura que en el caso anterior, nos moveremos hacia el objeto tanto como necesitemos y subiremos a la posición original.

En cualquier caso, la variable en cuestión en este ensayo es la altura que tenemos que desplazarnos en el eje Z para estar en el mismo plano con el actuador que nuestro objeto.

Una forma en la que se puede resolver este problema es realizando un barrido de la altura hasta que se detecte la deformación que se produce en la cerda al tocar el fondo sobre el que se sitúa el objeto. Otra posibilidad es el uso de algoritmos de visión 3D.

Por ejemplo, bajando iteración a iteración una determinada altura y tomando una imagen en cada instante, se puede identificar el actuador y obtener su tamaño y posición exactos. Incluso se puede identificar todos los puntos que forman parte de la misma, mediante las funciones de OpenCV que se han indicado en el ensayo anterior.

Comparando estos parámetros con unos obtenidos al comenzar el ensayo que sirvan de referencia es posible detectar variaciones y asociar una determinada diferencia entre parámetros a la deformación que buscamos. Guardando el valor del índice de la última iteración y conociendo cuánto se ha descendido en cada una, podemos conocer y guardar la altura del plano de trabajo con un sencillo producto.

3.3 ENFOQUE AUTOMÁTICO

Utilizando un algoritmo similar al del ensayo anterior se propone implementar un sistema de calibración que autoenfoque la imagen sin tener que intervenir para ello. La idea es realizar, de nuevo, un barrido de altura y, de la imagen tomada, calcular un valor que sea un índice del contraste. El valor de contraste más alto será el que corresponda con el mejor enfoque. Sabremos cual es el valor máximo hasta el que podemos iterar si previamente hemos realizado el ensayo de altura de trabajo que no requiere de enfoque ya que la cerda se encuentra siempre en la misma posición.

La forma de obtener el contraste es recorrer los puntos de la imagen en escala de grises comprobando su valor. Las diferencias entre puntos de objeto y puntos de fondo darán un índice del contraste que podremos simbolizar en un porcentaje. Tras el barrido el mayor contraste dará la solución.

4 RESULTADOS

Para la validación del proyecto se han llevado a cabo los ensayos propuestos en el punto 3 obteniéndose los siguientes resultados.

4.1 SEGUIMIENTO DE OBJETOS

Para el seguimiento de objetos se ha identificado una resistencia SMD como objetivo y se ha realizado el seguimiento de la misma hasta centrarla en la imagen tal y como se deseaba. El resultado del preprocesamiento se aprecia en la figura 7:

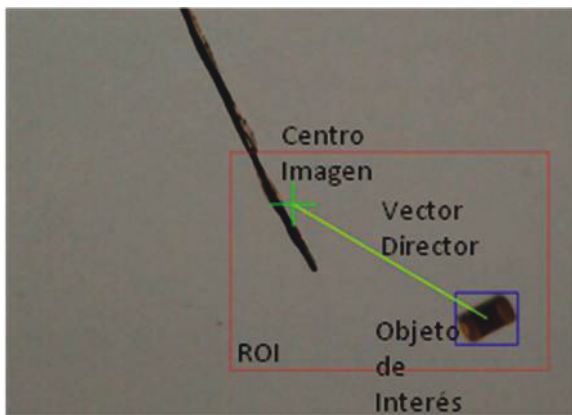


Figura 7: Resultado preprocesamiento para seguimiento

Podemos comprobar la correcta identificación del objetivo y la correcta definición de la región de interés con el tamaño adecuado. El proceso de seguimiento se desarrolla en las figura 8:

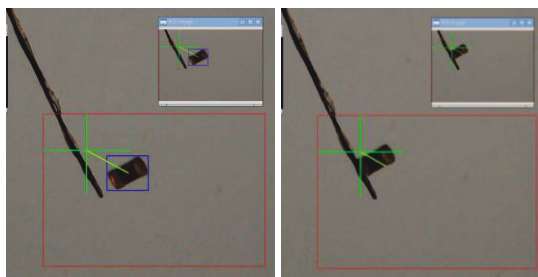


Figura 8: Proceso de seguimiento

Apreciamos como en dos instantes sucesivos, el sistema posiciona el objetivo. Si graficamos los datos del ensayo podemos ver la evolución de la posición en cada eje respecto a la referencia, por ejemplo, la figura 9 muestra el resultado para el eje Y:

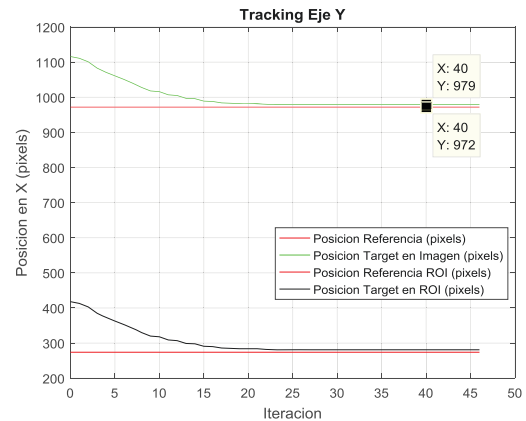


Figura 9: Resultado del seguimiento en el eje y

Pudiendo obtener los errores de posición en los ejes a partir de estos datos. Para este eje el error sería de 7 píxeles.

4.2 INTERACCIÓN CON OBJETOS

En cuanto a la detección de la deformación de nuestro actuador para la obtención de la altura de trabajo necesaria para la interacción con objetos, tenemos como resultado, para un primer método de comparación de tamaño y posición obtenidos por un rectángulo que delimita el objeto, los siguientes datos en la figura 10:

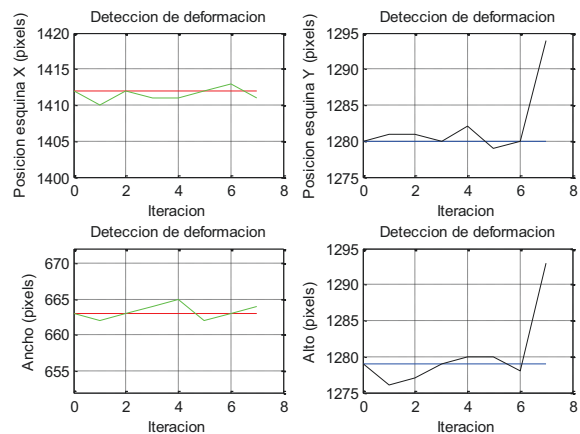


Figura 10: Detección de altura mediante el método del rectángulo

Donde apreciamos que la variación en la coordenada Y del punto inferior derecho del rectángulo y la variación en la altura del mismo debidos a la deformación permiten la detección después de haber realizado 7 iteraciones bajando una altura de 0.1mm en cada una de ellas.

Y para otra altura inicial, con un segundo método de comparación de los puntos del objeto con una tabla *Look-Up* construida con una imagen de referencia,

podemos comprobar la detección de puntos del contorno y el rectángulo del método anterior, así como los resultados de este segundo método en la figura 11.

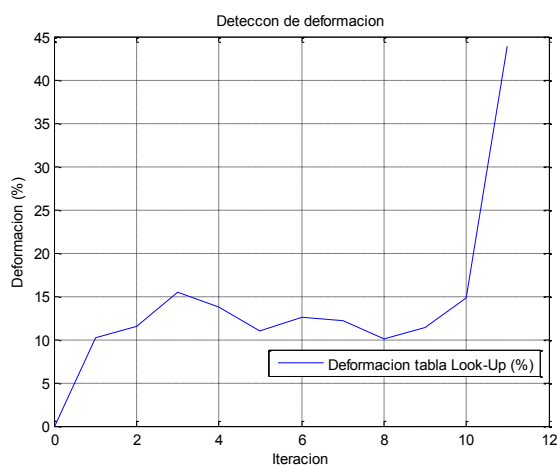


Figura 11: Calibración de altura mediante el método de comparación de puntos

Donde la deformación superior al 40% (valor límite tras varios ensayos) detectada, fija la altura del segundo ensayo a 1.1 mm después de 11 iteraciones bajando 0.1mm por iteración.

Tras la definición de esta altura de trabajo podemos proceder a tocar el objeto, mostrando el proceso la figura 12 que captura 2 instantes del mismo:

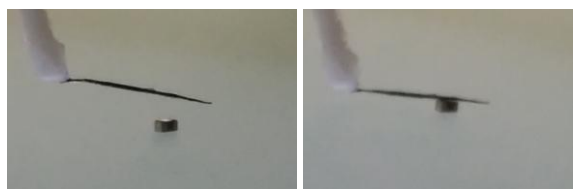


Figura 12: Proceso de tocar el objeto.

4.3 ENFOQUE AUTOMÁTICO

Por último, el resultado del ensayo de enfoque automático, siguiendo el procedimiento indicado se muestra en la figura 13.



Figura 13: Imagen antes y después de auto enfoque

5 CONCLUSIONES

5.1 SOBRE EL PROYECTO

A la vista de los resultados, queda patente la posibilidad de uso del prototipo con los fines para los que ha sido concebido. El control en bucle cerrado mediante visión por computador es solo una de las muchas posibilidades de proyectos y aplicaciones que pueden llevarse a cabo en este dispositivo. Las posibilidades de implementación y sucesivas mejoras solamente están limitadas al interés del usuario que, se espera, únicamente crezca mediante su uso.

5.2 FUTURAS MEJORAS E IMPLEMENTACIONES

Tras la implementación, desarrollo y ensayo con el prototipo se proponen multitud de cambios para mejorar la variedad y calidad de prácticas a implementar:

Una de estas mejoras consiste en ajustar el tamaño de la estructura del sistema al proyecto a implementar, lo que resulta sencillo gracias a la enorme modularidad del diseño. En el caso ensayado, el tamaño del sistema era superior al necesario debido a que la alta resolución de la cámara permite el trabajo con pequeños objetos y no se requieren grandes desplazamientos ni velocidades.

La sustitución de los motores paso a paso por servomotores de corriente continua aumentaría la precisión y la variedad en la implementación y control (lazo cerrado mediante *encoder*, nuevas etapas de potencia,...).

El cambio de diseño de la estructura a una con paneles abatibles que permitiera cerrar el área de trabajo permitiría un control de las condiciones de adquisición más preciso, mejorando esta notablemente al aislarse el sistema de la iluminación ambiente.

La posibilidad de modificar cualquiera de los componentes del sistema puede conducir a una mejora, como pudiera ser sustituir el Arduino de control por un kit de evaluación de STM (por ejemplo STM32F429-Discovery) que implemente un microcontrolador ARM con potencia de cálculo superior y posibilidad de tiempo real.

El uso de programas de diseño 3D y la producción mediante impresión 3D de elementos que puedan utilizarse para la realización de otros proyectos puede suponer, también, un acercamiento de las disciplinas de diseño y producción al alumno sirviendo esta excusa para el desarrollo de nuevas prácticas.

Agradecimientos

Este proyecto se ha realizado gracias a la beca de colaboración del Ministerio de Educación, Cultura y Deporte con la Universitat Politècnica de Valencia (PROYECTO: 2015/42/00005).

Referencias

- [1] A. Cardozo, "Motivación, aprendizaje y rendimiento académico en estudiantes del primer año universitario," *Laurus*, vol. 14, no. 28, pp. 209–237, 2008.
- [2] Brian Evans. *Practical 3D Printers: The Science and Art of 3D Printing*. Apress, Berkely, CA, USA, 1st edition, 2012. ISBN 1430243929, 9781430243922.
- [3] Benlloch, J. V., Agustí, M., Sanchez, A., & Rodas, A. (1995, October). Colour segmentation techniques for detecting weed patches in cereal crops. In *Proc. of Fourth Workshop on Robotics in Agriculture and the Food-Industry* (pp. 30-31).
- [4] Benlloch, J. V., Sanchez, A., Christensen, S., & Walger, M. (1996). Weed mapping in cereal crops using image analysis techniques. *AgEng96*, 2, 1059-1060.
- [5] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc., 2008.
- [6] G. Henderson, "Wiring Pi. GPIO Interface library for Raspberry Pi." .
- [7] J. Gómez Lahoz, N. Sánchez Martín, B. Arias Pérez, and D. González Aguilera, "Desarrollo de simuladores en investigación y docencia," presented at the TopCart 2004 Congreso Nacional de Topografía y Cartografía, 2004, pp. 59–60.
- [8] L. Greenwald, J. Kopena. Mobile robot labs. *Robotics. Automation Magazine, IEEE*, 10(2):25-32, 2003. ISSN 1070-9932.
- [9] M. Á. Cazorla Quevedo, V. Matellán Olivera, and J. M. Cañas, "Uso de simuladores en docencia de robotica movil," 2012.
- [10] P. Gil, J. Pomares, S. T. Puente Méndez, F. Torres Medina, F. A. Candelas-Herías, and F. G. Ortiz Zamora, "VISUAL : herramienta para la enseñanza práctica de la visión artificial," *Universitat Poiltècnica de Catalunya*, 2003.
- [11] Percoco, G., Lavecchia, F., & Salmerón, A. J. S. (2015). Preliminary study on the 3D digitization of millimeter scale products by means of photogrammetry. *Procedia CIRP*, 33, 257-262.
- [12] Percoco, G., & Salmerón, A. J. S. (2015). Photogrammetric measurement of 3D freeform millimetre-sized objects with micro features: an experimental validation of the close-range camera calibration model for narrow angles of view. *Measurement Science and Technology*, 26(9), 095203.
- [13] RepRap Community, "G-Code."
- [14] Ricolfe, C., & Sánchez, A. J. (2008). PROCEDIMIENTO COMPLETO PARA EL CALIBRADO DE CÁMARAS UTILIZANDO UNA PLANTILLA PLANA. *RIAI*, 5(1), 93-101.
- [15] Sanchez, A. J., & Martínez, J. M. (2000). Robot-arm pick and place behavior programming system using visual perception. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on* (Vol. 4, pp. 507-510). IEEE.
- [16] Sanchez-Salmeron, A. J., Lopez-Tarazon, R., Guzman-Diana, R., & Ricolfe-Viala, C. (2005). Recent development in micro-handling systems for micro-manufacturing. *Journal of materials processing technology*, 167(2), 499-507.
- [17] Sanchez-Salmeron, A. J., Lopez-Tarazon, R., Guzman-Diana, R., & Ricolfe-Viala, C. (2010). An inter-machine material handling system for micro-manufacturing based on using a standard carrier. *The International Journal of Advanced Manufacturing Technology*, 47(9-12), 937-943.
- [18] Sanchez-Salmeron, A. J., & Ricolfe-Viala, C. (2012). A flexible packaging station for micro-bulk-forming applications based on a standard carrier. *The International Journal of Advanced Manufacturing Technology*, 61(5-8), 529-536.
- [19] Sanchez-Salmeron, A. J., & Ricolfe-Viala, C. (2010). A bi-linear-vibratory feeder system for micro-bulk-forming applications. *STEEL RESEARCH INTERNATIONAL*, 81(9), 1205-1208.