

## SISTEMA DE VISIÓN OMNIDIRECCIONAL PARA ROBOTS MINI-HUMANOIDES

Félix Rodríguez Cañadillas (ferodrig@ing.uc3m.es)  
Martin F. Stoelen (mstoelen@ing.uc3m.es)  
Alberto Jardón Huete (ajardon@ing.uc3m.es)

### Resumen

*En este artículo se presenta el trabajo preliminar de un sistema de visión omnidireccional para robots mini-humanoides. Tales robots se encuentran actualmente en uso en varios centros de investigación de todo el mundo, y también son ampliamente utilizados en competiciones de robots, como por ejemplo, el concurso español CEABOT. Las plataformas robóticas mini-humanoides requieren de detección para realizar la interacción física con el entorno, aunque tienen una capacidad limitada de procesamiento a bordo de la información. El trabajo aquí presentado explora las diferentes maneras de utilizar la información de una cámara omnidireccional en un sistema robótico y qué ventajas se pueden obtener en el contexto de un concurso de robots mini-humanoides. La tarea de navegación de la competición CEABOT se utiliza como estudio de este caso. También se presentan las primeras ideas para la implementación, en concreto mediante la utilización una cámara Blackfin, con procesado a bordo, montada sobre un robot mini-humanoides Bioloid.*

**Palabras clave:** Robot mini-humanoides, cámara omnidireccional, CEABOT, detección de movimiento, flujo óptico, detección de obstáculos.

### 1. INTRODUCCIÓN

Uno de los principales desafíos en robótica es la comprensión del entorno mediante el uso de sensores, para darle al robot la capacidad de interactuar de forma autónoma.

Además es una manera de inspirar a jóvenes estudiantes de ingeniería a entrar en este campo a través de concursos de robótica, como se subrayó en [15] y [8]. Estos concursos son una manera fácil y activa de introducirse en el desarrollo de sistemas para el reconocimiento del entorno, además de una forma de comparar el desarrollo de nuestros sistemas con los del resto de participantes de estas competiciones. Un ejemplo es la competición española CEABOT [1], que se celebra anualmente, donde los equipos de estudiantes pue-

den probar sus robots mini-humanoides en cuatro pruebas diferentes. Esto se describe con más detalle en la sección II.

Los diferentes tipos de sensores que están disponibles para este tipo de competiciones, abren un amplio abanico de posibilidades de mejora para la investigación de esta área de la robótica mini-humanoides. Normalmente, los sensores usados en las competiciones de robots mini-humanoides están limitados por su cobertura del entorno que los rodea. Por ejemplo, los sensores infrarrojos que miden únicamente la distancia a un punto del entorno y por lo general requieren de 5 a 10 sensores para un reconocimiento aceptable, como podemos ver en [19].

El proyecto aquí descrito surgió de la motivación de crear un sistema de visión con una cobertura más completa para el robot. Este sistema debería mejorar las técnicas de interacción con el entorno que el robot mini-humanoides ya tiene, es decir, como complemento a los sensores de ultrasonidos, infrarrojos y presión.

Los sistemas de visión artificial son herramientas destacadas en las tareas de detección y reconocimiento en los sistemas robóticos. Entre las diferentes configuraciones usadas en robots mini-humanoides, podemos destacar dos: los sistemas de visión estéreo y los sistemas de visión omnidireccional. Entre los sistemas de estéreo, podemos destacar los utilizados por el robot mini-humanoides QRIO[17] y DARwinOP[4], que a través del procesamiento de la imagen estéreo es capaz de evitar obstáculos, como puede verse en [14]. Además vale la pena mencionar otros sistemas de visión estéreo que no están sujetos a plataformas comerciales. Por ejemplo, el desarrollado en [13], donde se puso en práctica un sistema de visión estéreo que se puede incorporar en cualquier plataforma mini-humanoides.

Los sistemas de visión omnidireccional son especialmente interesantes, ya que pueden obtener una vista de 360 grados del entorno. Para los sistemas de visión omnidireccional, existen varias configuraciones posibles, ver por ejemplo [16] y [11], donde a través de un reflector obtiene una visión

completa de todo el entorno. Esta configuración proporciona un resultado muy bueno, pero requiere un estudio de cómo crear este reflector, lo cual puede ser bastante complicado. Otra opción es adquirir un reflector comercial, aunque sus precios suelen ser elevados. Una opción alternativa es usar una lente de gran ángulo de visión, con la cámara apuntando hacia el suelo. Esto se describe con más detalle en la sección III.

En el resto de este artículo se describe el entorno de la competición CEABOT, seguido de las primeras ideas para la implementación de una cámara omnidireccional en el robot Bioloid, y una serie de sencillos algoritmos de visión por computador adecuados para tal sistema.

## 2. DESCRIPCIÓN DEL ENTORNO DE TRABAJO

Este proyecto se basa en la creación de un sistema de visión para un robot mini-humanoid que competirá en el concurso CEABOT. En [1], podemos encontrar más información de este campeonato, aunque aquí se describen las principales características de este campeonato.

Antes de elegir una plataforma de robot mini-humanoid en la que basar nuestro proyecto, debemos tener en cuenta las restricciones que se determinan en la normativa CEABOT, sobre todo la altura y las restricciones de peso. Estas son un peso máximo de 3 kg y una altura máxima de 50 cm con el robot completamente extendido. Además todo el cálculo computacional se debe realizar a bordo.

Observando de las plataformas del mercado, vemos que no todas las plataformas mini-humanoides cumplen con estos requisitos. Por ejemplo, plataformas como DARwinOP[4] y HOAP-3[9], que cuentan con el procesamiento de imágenes a bordo, no pueden participar en este torneo, ya que superan el peso y/o las limitaciones de altura. Hay otras plataformas en el mercado que cumplan con estos requisitos y que se utilizan habitualmente en los torneos de este tipo. Plataformas como Robonova[18] y Bioloid[3] son los más utilizados en ediciones anteriores de CEABOT. En el trabajo aquí presentado, se usó un robot Bioloid, ver figura 1.

El campeonato CEABOT tiene cuatro pruebas para determinar las capacidades de los robots mini-humanoides participantes. Estas pruebas son las siguientes:

- Carrera de obstáculos: Esta prueba se basa en la navegación por un entorno de obstáculos, teniendo que cruzar esta zona de un lado



Figura 1: Robot Bioloid.

a otro y regresar. El número de obstáculos utilizados puede variar durante la competición. Las líneas son de color amarillo, el suelo es de color verde y los obstáculos son de color blanco (ver figura 2), esto hace más fácil la detección de cada uno de los elementos del entorno.

- Escaleras: Esta prueba demuestra la capacidad del robot para subir escaleras. Una buena forma de apoyar el robot en la consecución de esta prueba será la localización de las escaleras, lo que puede determinarse mediante la detección de bordes.
- Sumo: En esta prueba, se realiza una lucha de Sumo entre dos oponentes robots. Una de las técnicas de visión por computador que pueden ser de utilidad aquí es la detección de movimiento, con la que podemos detectar si el robot oponente se mueve y, a qué velocidad y en qué dirección lo hace. Destacar que el área de la zona de combate está determinado por los reglamentos de CEABOT y marcado por una línea circular de color amarillo. Por tanto, es importante localizar esta línea con el fin de evitar el cruce de la misma.
- Exhibición libre: La prueba de exhibición se basa en una demostración libre de las habilidades del robot, por ejemplo la realización de una serie de movimientos coreográficos. Como se trata de una prueba en la que el robot normalmente no interactúa con el entorno, el uso de una cámara no es necesario. Sin embargo, si en esta demostración se utilizará un objeto externo, como una pelota o un cubo, una cámara podría ayudar al robot para interactuar con ese objeto.

A la vista de las pruebas anteriores y observando los sistemas robóticos utilizados en este tipo de torneo, se ha decidido centrarse en el desarrollo de un sistema de visión para apoyar la navegación en la prueba de carrera de obstáculos y para determinar la posición del oponente en la prueba de Sumo.

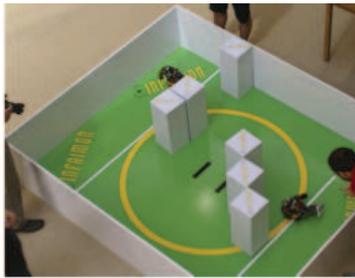


Figura 2: Entorno de trabajo de CEABOT durante la prueba de obstáculos.

### 3. SISTEMA DE VISIÓN OMNIDIRECCIONAL

Como se ha mencionado en la introducción, hay varias alternativas a tener en cuenta cuando se crea un sistema de visión omnidireccional. Uno es el que se propone en [8], donde a través de la reflexión de la imagen en un espejo cónico posicionado verticalmente en relación al robot es posible obtener una vista omnidireccional del entorno. Esta es una buena solución, pero requiere un espejo cónico, que puede ser bastante complicado y costoso de fabricar. Otra configuración posible es el uso de una lente con un amplio ángulo de visión, de modo que enfocando de la cámara hacia el suelo, desde la parte superior del robot, nos proporcione una vista omnidireccional del entorno, suficiente para observar objetos cercanos al robot.

Esta configuración omnidireccional es más sencilla de implementar, ya que sólo requiere de una estructura para separar el robot y la cámara, por ejemplo un tubo de plástico transparente. Para comprobar la viabilidad de esta configuración, se realizó un estudio del efecto de una lente de ángulo de visión amplio. Teniendo en cuenta que el robot mini-humanoide Bioloid tiene una altura de 31 cm, se determinó que la cámara se colocaría a 40 cm por encima del suelo, es decir, con una distancia de 9 cm entre la cámara y el robot. Esta altura proporcionaría una vista suficiente de los alrededores del robot, si por ejemplo utilizásemos una lente con un ángulo de visión de 120 grados. En la figura 3, 4 y 5, vemos el estudio de estas condiciones.

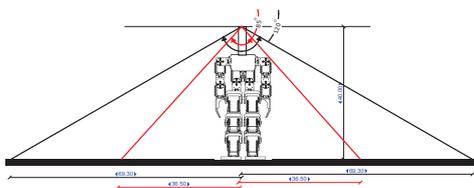


Figura 3: Vista frontal de la configuración omnidireccional.

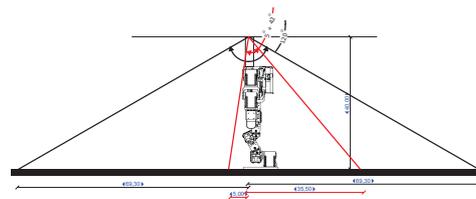


Figura 4: Vista de perfil de la configuración omnidireccional.

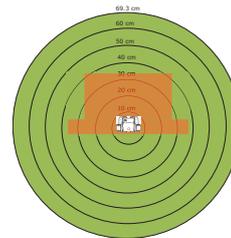


Figura 5: Vista de la planta de la configuración omnidireccional.

Para el trabajo aquí realizado se propone una cámara con procesamiento interno, para minimizar la carga computacional en el controlador del robot. Por ejemplo, la cámara Surveyor Blackfin [20], que puede obtenerse con una lente con un ángulo de visión de 120 grados. Esta cámara puede ser una buena opción para la implementación del sistema de visión a bordo de la plataforma Bioloid. Incorpora un procesador Blackfin BF537 de 500MHz con un sensor óptico de 1,3 megapíxeles. Además, incorpora una I/O externa de 32pins. Tiene una la masa total de 0.046 kg, y el tamaño es de 50x65x50 mm.

### 4. ALGORITMOS DE VISIÓN

En esta sección vamos a explicar los algoritmos de visión que se investigaron con el propósito de este proyecto. Todos estos algoritmos se han implementado en Matlab, ya que es una herramienta completa cuando se trabaja con el cálculo de matrices. Esto simplifica la creación de prototipos de los algoritmos, y permite evaluar y comparar estos algoritmos, antes de ponerlos en práctica en la propia cámara. Se supone que el robot está parado cuando se obtienen las imágenes.

#### 4.1. Detección Suelo-Obstáculo

Una de las principales características del entorno de trabajo de CEABOT es su simplicidad, ya que los diferentes componentes del entorno tienen diferentes colores. Si se captura una imagen en el entorno CEABOT desde una perspectiva omnidireccional, observamos que la imagen se compone de dos elementos definidos por su color, verde para

el suelo y blanco para los obstáculos. En la figura 6 podemos ver un ejemplo de esta imagen en el entorno real. La cámara omnidireccional utilizada en este ejemplo, es simulada por una cámara web, montada sobre una plataforma Bioloid.



Figura 6: Ejemplo de una imagen capturada en el entorno de CEABOT.

Una vez que la cámara se monta sobre el robot, y se mueve con él, calculamos la distancia desde el centro de la imagen hasta la frontera con las zonas blancas (obstáculos), de esta forma es posible estimar la distancia entre el robot y los diferentes obstáculos. El ángulo donde se localizan los obstáculos con respecto al robot también puede ser determinado fácilmente. A través del preprocesamiento de la imagen y su posterior binarización [10], se obtiene una imagen en la que estos cálculos pueden ser fácilmente llevados a cabo, como podemos ver en la figura 7.

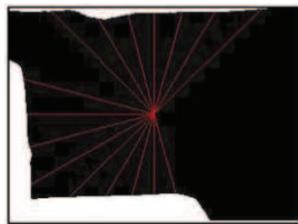


Figura 7: Representación de las distancias entre los obstáculos y el robot.

Para calcular las distancias entre el centro de la imagen y los obstáculos, hemos generado una serie de líneas con origen en el centro, por las cuales buscaremos el píxel de color blanco que nos determina la posición del obstáculo. El número de líneas que generamos depende de la cobertura que queremos o necesitamos para nuestro sistema. Para el ejemplo mostrado aquí, el reconocimiento de los obstáculos se realiza cada 15 grados, por tanto, debemos generar 24 líneas equidistantes. Si se cambia el número de líneas a generar, se debe calcular el ángulo de cada línea recta en respuesta a la siguiente ecuación:

$$Ang(n) = \frac{2\pi}{n_{total}} * n \quad (1)$$

Teniendo en cuenta la ecuación general de una recta  $y = ax + b$ , donde  $b$  es la  $y_0$  inicial, para nuestro caso 0,  $a$  es la pendiente de la recta trazada y definida como  $a = \tan(Ang(n))$ , podemos determinar que:

$$y = x * \tan(Ang(n)) \quad (2)$$

Con la ecuación (2), recorremos los valores de  $x$  por la línea recta del ángulo  $n$  para obtener un valor  $y$ . Habiendo calculado  $x$  e  $y$ , se comprueba el valor del  $pixel(x, y)$  en la imagen binaria. Si el valor de  $pixel(x, y)$  es igual a 0 significa que no hay obstáculos y por lo tanto no será necesario calcular la distancia, pero si el valor del  $pixel(x, y)$  es igual a 1 significa que hemos encontrado el obstáculo y por lo tanto ya no tendremos que continuar buscando a lo largo de la recta. Para calcular la distancia al objeto se calcula el modulo del vector generado por  $x$  e  $y$  y que podemos determinar como:

$$Dist(n) = \sqrt{x^2 + y^2} \quad (3)$$

donde  $Dist(n)$  es la distancia al obstáculo de la línea  $n$ . Repitiendo el proceso para todas las rectas se pueden obtener las distancias a los obstáculos en todo el entorno del robot. Este cálculo de todas las distancias se almacena en un vector  $Dist(n)$  que contienen todas las distancias calculadas.

Por último, debemos tener en cuenta que las distancias calculadas no son reales y representan la distancia en píxeles de la imagen. Para cambiar estos cálculos a las distancias del mundo real, se puede utilizar la proyección perspectiva [10] de la cámara, que sigue la siguiente ecuación:

$$x = f \frac{X}{Z} \quad y = f \frac{Y}{Z} \quad (4)$$

siendo las coordenadas en la imagen las variables  $(x, y)$  y las variables reales  $(X, Y)$ . Además,  $f$  es la distancia focal y  $Z$  es la altura a la cual está montada la cámara. Para obtenerlas coordenadas de píxel a partir de las coordenadas de la imagen, se puede utilizar los parámetros intrínsecos de la cámara, como se muestra en la siguiente ecuación:

$$x = -(x_{im} - o_x) * s_x \quad y = -(y_{im} - o_y) * s_y \quad (5)$$

donde  $(o_x, o_y)$  son las coordenadas del píxel central de la imagen, y  $(s_x, s_y)$  el tamaño de píxel en milímetros en la dirección horizontal y vertical de la imagen, respectivamente. Las distorsiones radiales están siendo ignoradas, pero deben ser incluidas en la aplicación final si son necesarias las distancias exactas.

Los resultados de este algoritmo son interesantes, ya que sus resultados parecen relativamente fiables en las imágenes del entorno real. Esto es, por supuesto, asumiendo que las condiciones de iluminación son relativamente constantes durante el

transcurso de la competición. Una posible mejora sería el uso de detección de bordes como un complemento a este algoritmo, aunque esto puede aumentar la carga computacional del sistema.

#### 4.2. Detección de movimiento

La estrategia que hemos utilizado para la detección del robot oponente en la prueba de Sumo se basa en detectar el movimiento de este robot. Como se muestra en el libro [10], cualquier movimiento perceptible de la escena puede ser calculado mediante una secuencia de imágenes, por lo que, si tales cambios son detectados, pueden analizarse las características de este movimiento. En el caso general del movimiento tridimensional de un objeto, sólo es posible obtener estimaciones cualitativas. De hecho, cuando un objeto se mueve en la dirección de su línea de observación, el movimiento no es perceptible, aunque existen algunas evidencias de dicho movimiento, como la variación de su tamaño en la imagen, el movimiento de la sombra, etc. Sin embargo, cuando el objeto se mueve en un plano perpendicular a la línea de observación, se pueden obtener buenas estimaciones de las componentes del movimiento.

Una teoría que se utiliza para la detección de movimiento es el uso de la imagen diferencia [10]. Esto se basa en la resta de imágenes, ya que se puede calcular la cantidad de movimiento de un único objeto como la diferencia entre dos imágenes de una secuencia. Por tanto, la imagen diferencia se define como  $Image_d$  y sigue la siguiente ecuación:

$$Image_d(p, t_1, t_2) = Image(p, t_2) - Image(p, t_1) \quad (6)$$

donde  $p = (x, y)$  es un píxel genérico de la imagen y  $t_1, t_2$  son los instantes de tiempo de dos imágenes consecutivas. Nótese que los valores o intensidades que se obtienen a través de (6) pueden ser negativos. El aspecto más atractivo de esta técnica es su simplicidad. Sin embargo, la información proporcionada no es muy descriptiva acerca de la forma de los objetos, aunque se puede detectar el área de la imagen donde se están produciendo estos cambios y concentrar el esfuerzo computacional en esta zona.

Para determinar la magnitud y la dirección del movimiento del robot oponente, podemos aplicar este algoritmo a tres imágenes consecutivas, de tal forma que obtendremos dos resultados, diferentes en el tiempo, de la imagen diferencia. Podemos ver una de estas imágenes diferencia en binario en la figura 8, en un ejemplo de una lata de refresco.

Al tener un único oponente y por lo tanto, un único elemento móvil, podemos calcular el centroide total de todas las regiones de captura de movi-



Figura 8: Ejemplo de la imagen diferencia en binario.

miento y suponer que pertenecen al mismo elemento móvil. Por tanto, podemos crear un vector entre los centroides de las dos imágenes diferencias obtenidas con anterioridad, de tal forma que este vector será representativo del movimiento del robot oponente. Para calcular el centroide de todas las regiones del movimiento captado en la imagen diferencia, la contribución de cada una de estas regiones ha de ser proporcional a la superficie de cada región dentro del conjunto total de movimiento capturado. Este algoritmo se realiza aplicando la siguiente ecuación:

$$Centroid_{total}(x, y) = \sum_{i=1}^n \frac{Area(i)}{Area_{total}} * Centroid(i) \quad (7)$$

Calculando estos dos centroides puede ser creado el vector que proporciona la magnitud y dirección de movimiento del objeto. Esto puede verse en la figura 9, donde el vector está superpuesto sobre la imagen original. Nótese que el vector de movimiento se encuentra solamente en el plano de la imagen, aunque aplicando este algoritmo sobre la configuración omnidireccional anteriormente descrita, no necesitaremos saber la profundidad a la que se encuentre el objeto a detectar.



Figura 9: Representación del vector velocidad en la imagen original.

Este algoritmo es una manera simple de calcular la posición del robot oponente cuando este se encuentra en movimiento. Debido a su simplicidad, no se ve muy afectado por la iluminación, por lo que es bastante robusto a diferentes condiciones de iluminación, siempre y cuando los cambios en la iluminación no sean demasiado pronunciados.

### 4.3. Centrado mediante flujo óptico

En esta sección se describe un algoritmo de visión implementado para apoyar a los algoritmos anteriormente descritos. Siguiendo la estrategia que proponen en [5], donde se utiliza el flujo óptico para centrar un robot móvil en tareas de navegación, podremos dar apoyo a nuestro algoritmo de detección suelo-obstáculo. Esto se basa en la comparación del flujo óptico máximo observado a ambos lados del campo visual, como un indicador de la proximidad a los obstáculos. Si el flujo óptico es mayor a la derecha, los objetos están más cercanos a este lado y, por tanto, el robot debe girar a la izquierda con el fin de evitar la colisión con el obstáculo.

El flujo óptico juega un papel importante en la estimación y descripción del movimiento, por lo cual es comúnmente utilizado en tareas de detección, segmentación y seguimiento de objetos móviles de una escena a partir de un conjunto de imágenes. Como observamos en [7], el flujo óptico puede ser descrito como un campo de vectorial sujeto a la condición de la ecuación de brillo constante y definido como el movimiento aparente del patrón de brillo de la imagen. Es decir, cualquier cambio en el brillo de una imagen que se produzca en un entorno de brillo constante debe de ser producido por el movimiento de algún elemento de la imagen. La ecuación de brillo constante se define como:

$$(\nabla E)^T * v + E_T = 0 \quad (8)$$

Como se indica en la ecuación, para una imagen  $E = E(x, y, t)$ , y un vector  $v$  de campo de movimiento, la suma del producto del gradiente de la imagen por el vector representativo del movimiento, más la imagen tomada dependiente del tiempo  $E_T$  ha de ser igual a cero para un entorno sin cambio de brillo, es decir, brillo constante. Por tanto, el flujo óptico es la aproximación del campo de movimiento, que puede ser calculado a partir de la captura de una secuencia de imágenes variables en el tiempo (ver figura 10), siempre y cuando asumamos las siguientes condiciones [7]:

- **Superficies Lambertianas:** Una superficie perfectamente difusora es aquella que emite o refleja la salida de luz de tal forma que presenta la misma luminosidad, independientemente del ángulo de visión. Tales superficies se denominan lambertianas ya que responden a la ley de Lambert.
- **Fuente puntual de luz en el infinito:** Se refiere a la fuente de luz, que ha de estar colocada a gran distancia de nuestro área de influencia.

- **Sin distorsión fotométrica:** La distorsión fotométrica representa la variación de contraste entre las imágenes, así que para el uso adecuado de flujo óptico la variación de contraste a lo largo del tiempo debe ser igual a cero.

Por lo tanto, para implementar nuestro algoritmo de flujo óptico debemos tener en cuenta los siguientes supuestos.

- La ecuación de brillo constante produce una buena aproximación de la componente normal del campo de movimiento.
- El campo de movimiento se aproxima bien al campo vector constante dentro de cualquier pequeña porción del plano de la imagen.

Tomando el primer supuesto, para cada punto de  $p_i$  en una pequeña región  $Q$ , de tamaño  $N \times N$ , se puede escribir como ecuación (8), donde las derivadas espaciales y temporales de brillo de la imagen se calcula en  $p_1, p_2, \dots, p_{N \times N}$ . Típicamente se utiliza una pequeña región de  $5 \times 5$ . Por lo tanto, el flujo óptico se puede estimar con  $Q$  como el vector constante,  $v$ , que minimiza la funcionalidad de la siguiente ecuación:

$$\Psi[v] = \sum_{p_{i \times j} \in Q} [((\nabla E)^T * v + E_T)]^2 \quad (9)$$

La solución a este problema de mínimos cuadrados puede ser determinada mediante el paso a un del sistema lineal, esto lo haremos de la siguiente manera:

$$A^T A * v = A^T * b \quad (10)$$

La fila  $i$  - *enesima* de la matriz  $A$  es el gradiente de la imagen espacial evaluada en el punto de  $p_i$ ,

$$A = \begin{pmatrix} \nabla E(p_1) \\ \nabla E(p_2) \\ \vdots \\ \nabla E(p_{N \times N}) \end{pmatrix} \quad (11)$$

y  $b$  es la dimensión  $N^2$  del vector de las derivadas parciales temporales de brillo de la imagen, evaluada en  $p_1, \dots, p_{N \times N}$ , después de un cambio de signo:

$$b = -[E_t(p_1), \dots, E_t(p_{N \times N})]^T \quad (12)$$

La solución de mínimos cuadrados del sistema a través de las restricciones de (10) puede obtenerse como:

$$v = (A^T A)^{-1} * A^T * b \quad (13)$$

donde  $v$  es el flujo óptico (el campo de movimiento estimado) en el centro de la región  $Q$ ; repitiendo este proceso para todos los puntos de la imagen, se obtiene un flujo óptico completo. Podemos ver una representación gráfica del resultado de este cálculo en la figura 11.



Figura 10: Entorno real de pruebas

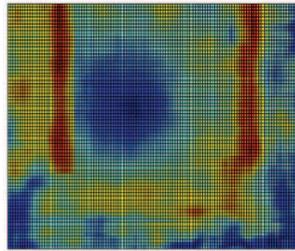


Figura 11: Representación gráfica del flujo óptico.

Para reducir la carga computacional de este algoritmo, hemos seguido una estrategia similar a la desarrollada en [12], enfocando la aplicación del algoritmo de flujo óptico en determinadas áreas de la imagen, en nuestro caso, en ambos lados del robot. La figura 12 muestra la representación del resultado del flujo óptico en las áreas más relevantes de nuestra imagen.

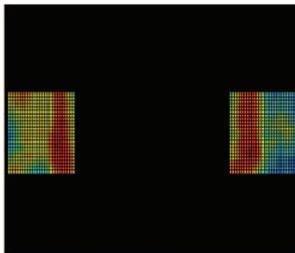


Figura 12: Representación gráfica del flujo óptico en las zonas de mayor relevancia.

Por último, cabe destacar que utilizando el algoritmo de flujo óptico podemos dar apoyo a los dos algoritmos anteriores, por lo que todo el sistema resulta más robusto y fiable. Nótese que este algoritmo es sensible a los cambios en la iluminación, aunque realizando un promediado del flujo óptico sobre áreas más relevantes puede obtenerse un resultado más robusto.

## 5. CONCLUSIONES

En el inicio de este proyecto nos hemos centrado en estudiar la normativa de CEABOT, las características requeridas por un sistema de visión omnidireccional y las características de los robots

mini-humanoides. A través de estos tres estudios, se han obtenido los factores determinantes en el desarrollo de este proyecto, como por ejemplo la altura del robot con la cámara y las dimensiones del entorno CEABOT. A partir de la elección de la cámara con una lente de 120 grados de ángulo de visión y montada a 40 cm del suelo, hemos podido comenzar el desarrollo de los algoritmos de visión.

En cuanto a los tres algoritmos de visión, los prototipos iniciales que hasta ahora se han implementado en Matlab y probado con imágenes del entorno real de CEABOT, nos han reportado unos buenos resultados. Los tres son relativamente fáciles de aplicar y tienen un requisito computacional bajo. Sobre todo el de detección de movimiento, que será nuestra referencia para la prueba de Sumo. Para la prueba de carrera de obstáculos probablemente podamos beneficiarnos de los resultados del algoritmo de detección de suelo-obstáculo, mientras que el algoritmo de flujo óptico puede requerir más trabajo para ser útil.

En general, la amplia vista proporcionada por una cámara omnidireccional proporciona un aumento significativo en capacidad de detección del entorno, y por tanto podremos dar apoyo a los sensores comúnmente utilizados en los robots mini-humanoides.

## Agradecimientos

Los autores de este trabajo quieren agradecer el apoyo de COMANDER CCG10-UC3M/DPI-5350 y de ROBOCITY2030, proyectos financiados por la Comunidad de Madrid y la UC3M. Gracias a la Asociación de Robótica de la Universidad Carlos III (ASROB) [2], por darnos los medios para desarrollarlo.

## Referencias

- [1] A. Jardón, P. J. Sanz, F. Gómez, J. Felip and J. C. García. CEABOT'10 - Normativa. Comité Español de Automática (CEA), Universidad de Jaén. Spain, 2010.
- [2] Asociación de Robótica de la Universidad Carlos III (ASROB), 2012. [Online]. Avaliable: <http://asrob.uc3m.es>
- [3] Bioloid, ROBOTIS, 2012. [Online] Avaliable: [robotis.com/xen/bioloid-en](http://robotis.com/xen/bioloid-en)
- [4] DARwinOP, Open Platform Humanoid Robot for Research and Education, 2012. [Online]. Avaliable: [romela.org](http://romela.org)
- [5] D. Cooms and K. Roberts, Centering behavior using peripheral vision, Computer Vision

- and Pattern Recognition. Proceedings CVPR 93, IEEE Computer Society Conference, New York, 1993. pp. 440-445.
- [6] E. Menegatti, E. Pagello and M. Wright, Using Omnidirectional Vision within the Spatial Semantic Hierarchy. Conference on Robotics and Automation. Proceedings. ICRA 02. IEEE International, vol.1. 2002. pp. 908-914.
- [7] E. Trucco and A. Verri, Introductory Techniques for 3-D Computer Vision. New Jersey: Prentice Hall, 1998, ch. 8.
- [8] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda and E. Osawa. RoboCup: The Robot World Cup Initiative. IJCAI-95 Workshop on Entertainment and AI/Alife, 1995.
- [9] HOAP-3, Miniature Humanoid Robot. Fujitsu, 2012. [Online] Available: [home.comcast.net/jtechsc/HOAP-3-SpecSheet.pdf](http://home.comcast.net/jtechsc/HOAP-3-SpecSheet.pdf)
- [10] Javier González Jiménez, Visión por computador. Madrid, PARANINFO, 1999.
- [11] J. Kim and Y. Suga. An Omnidirectional Vision-Based Moving Obstacle Detection in Mobile Robot. International Journal of Control, Automation and Systems, vol. 5, no. 6, 2007. pp. 663-673.
- [12] J. van der Blij, Omnidirectional Active Vision in Evolutionary Car Driving, M.Sc. Graduation Thesis in Artificial Intelligence, University of Groningen (RuG), Groningen, The Netherlands, 2005.
- [13] K. Konolige. Small Vision Systems: Hardware and Implementation. In Eighth International Symposium on Robotics Research. Hayama, Japan, 1997.
- [14] K. Sabe, M. Fukuchi, J.S. Gutmann, T. Ohashi, K. Kawamoto and T. Yoshigahara. Obstacle avoidance and path planing for humanoid robots using stereo vision. In Int. Conf. on Robotics and Automation (ICRA), Seoul, Korea, 2001.
- [15] M. González, A. Jardón, S. Martínez, M.F. Stoelen, J.G. Victores and C. Balaguer. Educational initiatives related with the CEABOT contest. In Proc. of 2nd Int. Conf. on Simulation, Modeling and Programming for Autonomous Robots (SIMPAN 2010), Darmstadt, Germany, pp. 649-658.
- [16] N. Winters, J. Gaspar, G. Lacey and J. Santos. Omni-directional Vision for Robot Navigation. IEEE Workshop on Omnidirectional Vision, 2000.
- [17] QRIO, Sony Dream Robot, 2012. [Online]. Available: <http://sonyaibo.net/aboutqrio.htm>
- [18] Robonova, Hitec Robotics, 2012. [Online] Available: [robonova.de/store/home.php](http://robonova.de/store/home.php)
- [19] S. Pons, N. Tolós and S. Troyano. Algorismes i modifications hardware a la robot Dorami per participar al campionat nacional de robotica humanoide CEABOT 2010. Technical Report, Institut de Robotica i Informatica Industrial, 2010.
- [20] Surveyor SRV-1 Blackfin Camera, Surveyor Corporation, 2012. [Online] Available: [surveyor.com/blackfin/](http://surveyor.com/blackfin/)