

Detección de la Orientación mediante Visión Artificial para el Control de Equilibrio en Robots Humanoides

Juan Miguel Garcia-Haro¹, Santiago Martínez¹, Carlos Balaguer¹

¹RoboticsLab, Universidad Carlos III de Madrid, Avda. Universidad 30, Leganés (Madrid), España, {jgarciah, scasa, balaguer}@ing.uc3m.es

Resumen

Los métodos principales para controlar el equilibrio de robots humanoides están basados principalmente en la utilización de indicadores como el “Foot Rotation Point” (FRI), o el “Zero Moment Point” (ZMP). Estos métodos están basados en un control dinámico del equilibrio. Sin embargo, en situaciones estáticas, la visión periférica es una de las principales herramientas de control del equilibrio en seres humanos. Por lo tanto, este proyecto se centrará principalmente en el estudio de los distintos algoritmos que permitan calcular el grado de inclinación del robot, y a futuro, realizar un control de equilibrio estático del robot inspirándose en los seres humanos. En este caso, utilizando la visión artificial del robot a través de imágenes RGB.

Palabras clave: Visión, Estabilidad, Control, Humanoide.

1. INTRODUCCIÓN

El control de equilibrio en robots humanoides es un tema que lleva desarrollándose desde muchos años. Diferentes aplicaciones, métodos o algoritmos se han ido desarrollando paulatinamente para mejorar el comportamiento de este tipo de robots ante diferentes perturbaciones.

Un ejemplo de estudio de control de estabilidad podría ser el un robot humanoide camarero, como el que se está llevando a cabo por el grupo de investigación “RoboticsLab” en la Universidad Carlos III de Madrid mediante el robot TEO (Task Environment Operator) (Figura 1), Para controlar la estabilidad de la bandeja [7] o del propio cuerpo del robot [13], se pueden aplicar distintos métodos, en los que uno o varios sistemas sensoriales puede utilizarse para cumplir con la tarea.

De hecho, algunos de estos sistemas de control de equilibrio están bio-inspirados, como puede ser en el sistema de vestibular. Éste está formado por órganos internos del oído, y es el principal encargado de ayudarnos a mantener el equilibrio y conocer la orientación y posición de nuestra cabeza.

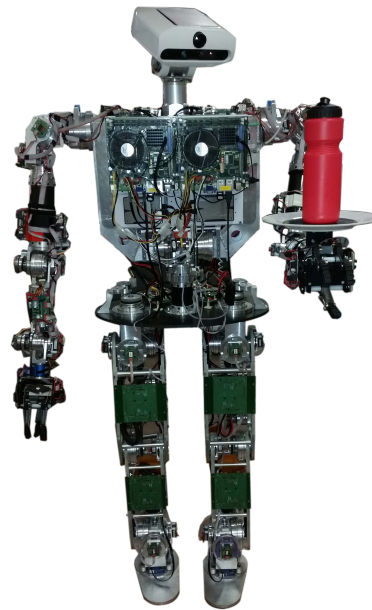


Figura 1: El robot humanoide TEO desarrollado en la Universidad Carlos III por el grupo de investigación “RoboticsLab”, trabajando como camarero.

Sin embargo el sistema vestibular no es el único sistema involucrado en el control del equilibrio. La visión es también una herramienta muy importante para el control del estabilidad. Se han realizado estudios que demuestran en base a personas mayores que una pérdida de visión afecta significativamente a la capacidad de mantener el equilibrio y aumenta el riesgo de caída [8, 9, 3].

Por tanto resulta bastante interesante utilizar la visión artificial como posible herramienta para el control de la postura y el equilibrio en robots humanoides. Para este artículo, se estudiará distintos enfoques para conocer la orientación del robot (el grado de desequilibrio), profundizando principalmente en el métodos basados en “Perspective-n-Point” (PnP). Y más concretamente, experimentando con el método iterativo.

También se analizarán las posibilidades de poder trabajar sin necesidad de tener una referencia concreta del mundo, es decir, de poder realizar el control de estabilidad del robot en cualquier entorno.

Para facilitar la lectura, este artículo está organizado de la siguiente manera: el siguiente capítulo presenta los problemas que pueden surgir en el control de equilibrio y cómo el uso de información visual a través de una cámara juntos a otros sensores puede ayudarnos en la tarea. En el tercero, se introduce el estado del arte sobre métodos PnP para la obtención de las variables necesarias para un futuro control. Y el cuarto presenta los experimentos y resultados obtenidos según los distintos métodos aplicados. En el último capítulo, se explican unas breves conclusiones y también unos futuros trabajos para integrar la fusión sensorial de fuerza/par e inercial.

2. EXPOSICIÓN DEL PROBLEMA

El control de estabilidad de robots humanoides es una multitarea, en la que múltiples sensores proveen de información al sistema de control. Es muy importante que la información transmitida por cualquier sensor sea lo más clara y precisa posible. De esta manera, facilitaremos que el control sea más robusto y tengamos menos problemas para completar cualquier objetivo.

Pero es inevitable que los sensores aporten información errónea o por lo menos no suficientemente precisa. El robot contiene holguras mecánicas, flexibilidad, etc., que sumadas al posible ruido introducido a los sensores, hace que la información no sea adecuada.

Sin embargo, existe la posibilidad de fusionar los datos de cada sensor. La capacidad para generar un valor más robusto con la comparación de información que provenga de distintos lugares hará que el control funcione mejor.

Para empezar, los enfoques principales para mantener el equilibrio de robots humanoides han sido basados principalmente en la utilización de indicadores de estabilidad como FRI (Foot Rotation Point) [5], ZMP (Zero Moment Point) [14], en la velocidad de variación del momento angular [6] o en Redes neuronales [10].

Estos métodos están basados en un control dinámico del equilibrio y utilizan principalmente la superficie apoyada del pie como referencia del equilibrio del robot. Para desarrollar el controlador de estabilidad, se utilizan sensores de fuerza/par o inerciales que son capaces de medir las fuerzas que se generan en el cuerpo del robot.

Sin embargo, para una situación estática, la visión periférica es una de las principales herramientas de mantener la estabilidad para el caso de seres humanos [4, 2].

Aprovechando esta bio-inspiración, este trabajo se centrará principalmente en estudiar una forma de controlar el equilibrio estático del robot inspirándose en la visión foveal o periférica de los humanos. De esta manera, en un trabajo futuro se podrá implementar un sistema que fusione los datos visuales con otros sensores como los de fuerza/par o inerciales.

3. ESTADO DEL ARTE

Para conocer las rotaciones u orientaciones del robot existen diferentes enfoques, como algoritmos de transformación lineal directa o algoritmos de filtro de partículas, aun que este artículo se centra en los mencionados previamente “Perspective-n-Point”. La razón principal está relacionado con el “time complexity”. Para los métodos PnP, el orden de estos algoritmos es más bajo. Esta condición es necesaria para un control de estabilidad, pues necesitaremos de que la información sensorial refleje rápidamente cualquier perturbación sobre el robot.

Principalmente, estos algoritmos PnP se basan en conocer n puntos en 3D con respecto al sistema de referencia del mundo (X,Y,Z) , así como sus respectivas proyecciones 2D en el plano de la imagen (u,v) . Conociendo los parámetros intrínsecos de la cámara (distancia focal, distorsión, centro de la imagen), se podrán calcular las matrices de rotación del sistema de referencia del mundo con respecto al de la cámara o robot, y así obtener los ángulos de Euler (Guiñada, Alabeo y Cabeceo) [1, 15] (Figura 2).

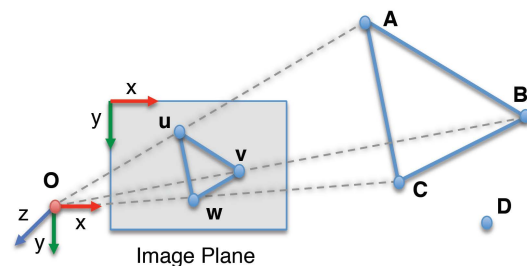


Figura 2: Representación básica del problema PnP, donde el triángulo ABC es proyectado sobre el plano de la imagen definiendo por tanto un triángulo proyectado uvw .

Los puntos expresados en el sistema de coordenadas *mundo* se proyectan en el plano de la imagen (u_1, u_2) usando la matriz Π (modelo de proyección en perspectiva) y la matriz A (parámetros intrínsecos de la cámara):

$$\begin{bmatrix} u_1 \\ u_2 \\ 1 \end{bmatrix} = A \Pi M_w^c \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

$$A = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \Pi = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Y M_w^c , la “pose” estimada, son los vectores de rotación y traslación que permiten transformar un punto 3D expresado desde el sistema de coordenadas *world* al sistema de la *cámara*.

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = M_w^c \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

3.1. Métodos No-Iterativos

Existen diferentes algoritmo no-iterativos que resuelve con mejor o peor resultado. Uno de los más conocidos es el método P3P introducido por Xiao-Shan Gao [16], “Complete Solution Classification for the Perspective-Three-Point Problem”. En este caso, para obtener la “pose”, la función requiere exactamente de cuatro puntos y una imagen de puntos (sus proyecciones). Los primeros 3 puntos se usan para estimar todas las soluciones posibles del problema P3P, el último se usa para conseguir la mejor solución que minimiza el error de proyección.

Otro algoritmo es el método EPnP introducido por F.Moreno [11], “EPnP: An Accurate Non-Iterative O(n) Solution to the PnP Problem”. Éste es un método que parte de 4 puntos o más y expresa los n puntos 3D como una suma ponderada de 4 puntos de control, con los cuales calcula las matrices de rotación. Es un método interesante debido a que es más robusto que el método P3P, pero computacionalmente más complejo.

3.2. Métodos Iterativos

El método iterativo se basa en la optimización de Levenberg-Marquardt. También conocido como el método de mínimos cuadrados amortiguados (Damped Least-Squares DLS), es una técnica iterativa que localiza el mínimo de una función multi-variable que se expresa como la suma de cuadrados de funciones de valores reales no lineales [12]. Se ha convertido en una técnica estándar para problemas de mínimos cuadrados no lineales, ampliamente adoptada en un amplio espectro de disciplinas, como en este caso visión.

En este caso, la función se centra resolver el problema PnP y encontrar la “pose” mientras que a su vez minimiza el error de proyección, el cual es la suma cuadrática de las distancias entre las proyecciones observadas de los puntos y los dichos puntos proyectados. El número mínimo de puntos es 3 (3 puntos son suficientes para calcular una “pose”, pero hay hasta 4 soluciones posibles). La solución inicial debe estar cerca de la solución global para converger como condición necesaria. Es un método bastante robusto al igual que el EPnP, pero pero computacionalmente más sencillo y si orden O(n) es más bajo. Y por eso es el método que se ha desarrollado finalmente.

4. EXPERIMENTOS Y RESULTADOS

Como se mencionó previamente, el objetivo de este proyecto es buscar una forma de imitar el control del equilibrio en situaciones estáticas de los seres humanos aplicándolo a un robot humanoide, el cual se basa principalmente en sensores propioceptivos como los sensores de fuerza/par, inerciales, o cámaras de visión.

Para ello, se ha procedido a realizar dos tipos de experimentos. Los primeros test se basan en buscar una referencia en el entorno para conocer la posición relativa de la cámara con respecto al mismo (o como en aviónica se llama “falso horizonte”).

Los segundos test se centran en la aplicación de métodos PnP iterativos para obtener la “pose” de un objeto a través de la correspondencia 3D-2D de los puntos de éste.

4.1. Línea de Falso Horizonte

Primero se pretende imitar el funcionamiento del conocido como PVHD (Peripheral Vision Horizon Display), un falso horizonte que se utiliza en las aeronaves para ayudar al piloto a conocer en todo momento la orientación del vehículo en el vuelo en todo momento (ángulos de guiñada, cabeceo y alabeo del avión).

Para esto se utilizó el método iterativo de Lucas-Kanade con pirámides, el cual nos permite hacer un seguimiento de un punto concreto de la imagen. Para nuestro caso, buscando dos puntos en el centro de la imagen y calculando el optical flow de estos dos, se puede seguir su movimiento, y de esa manera, conocer la orientación de la cámara.

En las Figuras 3 y 4 se puede observar como el paso inicial es escoger dos puntos a la misma altura en píxeles para poder generar el horizonte virtual o falso horizonte. Una vez se han fijado, el algoritmo

detecta como se van desplazando esos dos puntos en función de la rotación de la cámara, la cual se obtiene como resultado.



Figura 3: Test 1. Generación de la línea de horizonte utilizando el método de Lucas-Kanade.



Figura 4: Test 1. Comprobación de grado de inclinación la línea de horizonte rotando la cámara.

A partir de la misma idea del horizonte falso, se desarrolla un segundo programa que en lugar de simular un horizonte ficticio o virtual, detecta líneas en el entorno mediante el método HoughLines de detección de bordes.

Esta evolución implica no centrarnos en una línea ficticia, sino que aprovechamos el entorno para saber cómo de estables somos, o en este caso el robot.

Una vez aplicado el algoritmo de detección de bordes y fijada la línea que usamos como referencia, y otra vez, basándonos en el método del cálculo del optical flow de dos puntos de la línea (que se utilizó para el horizonte virtual), se hace un seguimiento del movimiento de dicha línea.

La ventaja radica en que ahora, al mismo tiempo se utiliza la posición original de la línea como Homing de referencia para calcular el ángulo de rotación de la línea cuyo movimiento se sigue (Figuras 5 y 6).

Aunque ambos algoritmos funcionaron, finalmente estas ideas fueron desechadas debido a varios motivos: el método de Lucas-Kanade no era lo suficientemente robusto como para considerarlo una herramienta válida para conocer el equilibrio del robot. El movimiento o giro, a veces rápido, de la

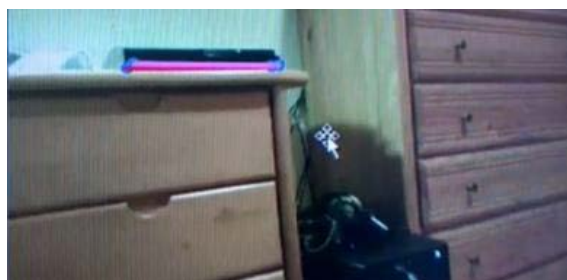


Figura 5: Test 2. Generación de la línea de horizonte basado en detección de bordes del entorno.



Figura 6: Test 2. Comprobación de grado de inclinación la línea de horizonte rotando la cámara.

cámara hacía que el algoritmo de detección se perdiera. En muchas ocasiones, la línea de referencia no correspondía correctamente al movimiento de la imagen,

Lo mismo sucedía con el algoritmo de detección de bordes. Los giros de la cámara provocaban que la línea de horizonte falso o Homing de referencia no se moviese a la par. De tal manera que el Homing de referencia se iba desplazando por la imagen, sin mantener la propiedad de Homing.

A esto hay que añadirle que, en ambos algoritmos, estas líneas eran simples representaciones en 2D de una línea del entorno, por lo que el cálculo de los ángulos de rotación tendría muchos errores relativos a la perspectiva.

4.2. Método Iterativo

Finalmente se ha decidido utilizar los métodos basados en PnP, los cuales ofrecen una solución realmente válida el problema que se presenta, ya que utilizaba la proyección de puntos 3D en el plano de la imagen pudiendo obtener la orientación de la cámara con 6 grados de libertad. Todo esto de una manera mucho más robusta, sin errores relacionados con el método de Lucas-Kanade o de HoughLines.

De los diferentes métodos de resolver PnP que se expusieron previamente en este artículo, se decide implementar el modelo iterativo, el cual ofrece una menor cantidad de error que los otros restan-

tes para el caso de trabajar con el típico tablero de damas o damero como referencia. El algoritmo funciona siguiendo una secuencia que consiste en: primero aplicar el método iterativo sobre la imagen de la cámara y 3 puntos (en este caso, 3 esquinas de los cuadrados del damero) de ésta. Segundo calcular los vectores de rotación. Tercero, transformar los vectores de rotación en matrices de rotación y traslación. Por último, una vez se obtienen las matrices mencionadas, se aplica una descomposición de dichas matrices de rotación y traslación (matriz de proyección), para obtener una matriz de calibración y rotación y la posición de una cámara. O dicho de otra manera, obtener tres ángulos de Euler que nos informan sobre la rotación que se produce en los tres ejes del sistema de referencia de la cámara.

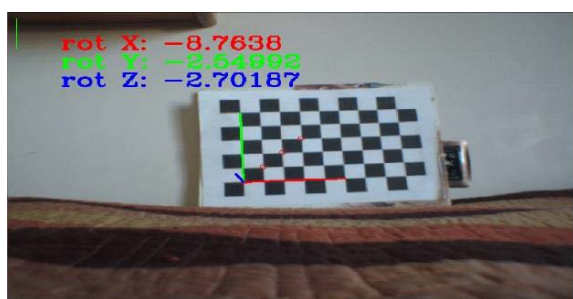


Figura 7: Test 3. Comprobación de la rotación para un giro negativo sobre el eje de rotación X (eje rojo).

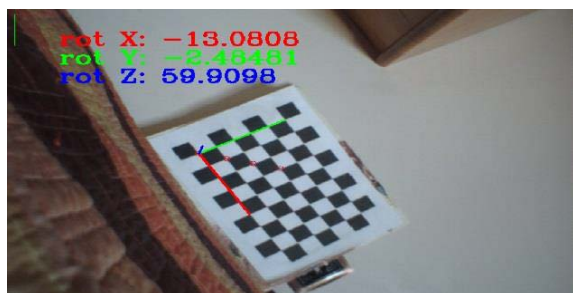


Figura 8: Test 3. Comprobación de la rotación para un giro positivo sobre el eje de rotación Z (eje azul).

Como se puede ver en las Figuras 7 y 8, los resultados son bastante alentadores. Y para corroborarlos se ha realizado un test adicional comparando el valor de los ángulos estimados obtenidos respecto a los valores de giro de un sensor inercial colocado en el centro de gravedad del robot. El test ha consistido en ir moviendo el robot en variaciones de dos grados (ang_{ref}) tanto en el plano frontal como en el plano sagital para ver la evolución de los ángulos ambos sensores, la cámara (Y_{cam} , Z_{cam}) y el sensor inercial (Y_{imu} , Z_{imu}). En las Figuras 9 y 10 se muestran estos resultados.

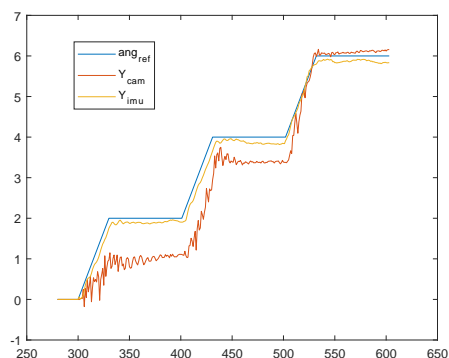


Figura 9: Test 3. Comprobación de la rotación para un giro negativo sobre el eje de rotación X (eje rojo).

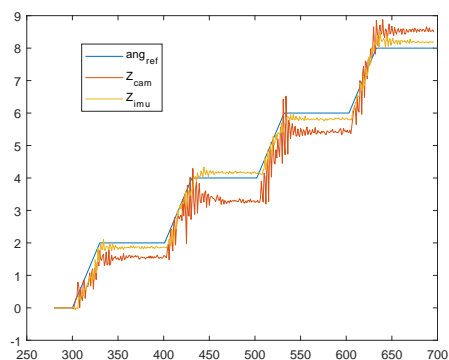


Figura 10: Test 3. Comprobación de la rotación para un giro positivo sobre el eje de rotación Z (eje azul).

En el plano del “frame” de la imagen, tanto los giros grandes en el eje X (eje rojo), como los giros grandes en el eje Y (eje verde), están bien posicionados y el algoritmo responde de manera correcta. Los valores obtenidos son precisos (el error medio es de 3.2%) y no se comprueba ningún efecto negativo antes de giros bruscos de la cámara.

Para el eje de rotación Z (eje azul), los resultados son aún mejores. En primer lugar, tampoco se observa ningún efecto perjudicial en el cálculo de este ángulo ante giros de la cámara bruscos. Y en segundo lugar, la disposición perpendicular de este eje al plano de la imagen hace más fácil la obtención del valor de giro, de tal manera que el error medio para este eje se reduce hasta el 1.4%.

Sin embargo, cuando nos acercamos a valores más pequeños, que están próximos al cero, el error aumenta entorno al 20% y en algún caso, incluso hasta el 50%. Además hay que añadir, como aporte negativo, el ruido que se acompaña a la información visual cuando se está moviendo el robot.

La razón de investigar en ángulo cerca del cero se debe a que el rango de equilibrio del robot estará dentro de estos límites. Los tests ante ángulos negativos han sido iguales a los mostrados en las Figuras 9 y 10 y por eso no se muestran.

5. CONCLUSIONES

Este artículo asienta las bases para en un futuro realizar un control de estabilidad mediante el uso de visión artificial. Se ha visto como algunos métodos que trabajan en el plano de la imagen 2D funcionan bien, aun que no con la suficiente robustez. Por eso, se pasó a utilizar métodos relacionados con el problema PnP.

Como primera aproximación, de un algoritmo joven y aun por mejorar, podemos concluir que cumple con los objetivos propuestos de obtener los ángulos de Euler, y así proporcionar un posible input al controlador de equilibrio del robot. Aun que los valores obtenidos no son muy precisos dentro de los rangos óptimos para el equilibrio del robot.

Como trabajos futuros, se han marcado varios objetivos tanto a corto como a largo plazo. A corto plazo y en primer lugar se va a pretender buscar formas de conocer la rotación de la cámara sin necesidad de un objeto conocido (como es en este caso el damero) para poder utilizar este método en cualquier tipo de entorno (como podría ser a través de mapas 3D). En segundo lugar, si no fuera eficiente el anterior punto expuesto, se vería utilizar como referencia del mundo una pared, suelo del laboratorio u otro objeto o superficie. Como requisito indispensable, estos objetos o superficies deberían ser ortogonales para facilitar los cálculos.

A largo plazo podríamos desarrollar dos tareas más. Una sería la de implementar el algoritmo de visión artificial desarrollado en el robot humanoide TEO y crear el programa de control para poder mover el robot humanoide hacia posiciones de equilibrio. Aquí habrá que tener en cuenta si el cuello de robot estará fijo o no, para determinar el grado de estabilidad del robot. Y otra sería la de intentar fusionar la información sensorial de la cámara con otros sensores relacionados con el equilibrio del robot, como el inercial.

Agradecimientos

La investigación que lleva a estos resultados ha recibido financiación del proyecto ROBOHEALTH-A (DPI2013-47944-C4-1-R), financiado por el Ministerio de Economía y Competitividad español y del proyecto RoboCity2030-III-CM (S2013 / MIT-2748) , financiado por Programas de Actividades de I + D en la Comunidad de Madrid y cofinanciado por los Fondos Estructurales de la UE.

English summary

Pose Estimation by Artificial Vision for Balance Control in Humanoid Robots

Abstract

The main methods to control the balance of humanoid robots are based on the use of indicators such as the Foot Rotation Point (FRI), or the Zero Moment Point (ZMP). These methods are based on a dynamic balance control. However, in static situations, peripheral vision is one of the main tools of balance control in human beings. Therefore, this project will focus mainly on the study of an algorithm that allows to calculate the estimated pose of the robot, and then, the performance of a bio-inspired static balance control of the robot. In this case, using the artificial vision of the robot through a kinect camera.

Keywords: Vision, Stability, Control, Humanoid.

Referencias

- [1] A. Ansar and K. Daniilidis. Linear pose estimation from points or lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):578–589, may 2003.
- [2] A. Berencsi, M. Ishihara, and K. Imanaka. The functional role of central and peripheral vision in the control of posture. *Human Movement Science*, 24(5-6):689–709, oct 2005.
- [3] A. L. Coleman, K. Stone, S. K. Ewing, M. Nevitt, S. Cummings, J. A. Cauley, K. E. Ensrud, E. L. Harris, M. C. Hochberg, and C. M. Mangione. Higher risk of multiple falls among elderly women who lose visual acuity. *Ophthalmology*, 111(5):857–62, may 2004.
- [4] J. Dickinson and J. A. Leonard. The Role of Peripheral Vision in Static Balancing. *Ergonomics*, 10(4):421–429, jul 1967.
- [5] A. Goswami. Postural Stability of Biped Robots and the Foot-Rotation Indicator (FRI) Point. *The International Journal of Robotics Research*, 18(6):523–533, jun 1999.

- [6] A. Goswami and V. Kalle. Rate of change of angular momentum and balance maintenance of biped robots. In *IEEE International Conference on Robotics and Automation*, pages 3785–3790 Vol.4. IEEE, 2004.
- [7] J. Hernandez-Vicen, S. Martinez, J. M. Garcia-Haro, and C. Balaguer. Correction of Visual Perception Based on Neuro-Fuzzy Learning for the Humanoid Robot TEO. *Sensors*, 18(4):972, mar 2018.
- [8] T. Hong, P. Mitchell, G. Burlutsky, C. Samarawickrama, and J. J. Wang. Visual Impairment and the Incidence of Falls and Fractures Among Older People: Longitudinal Findings From the Blue Mountains Eye Study. *Investigative Ophthalmology & Visual Science*, 55(11):7589, nov 2014.
- [9] B. E. Klein, R. Klein, K. E. Lee, and K. J. Cruickshanks. Performance-based and self-assessed measures of visual function as related to history of falls, hip fractures, and measured gait time: The beaver dam eye study. *Ophthalmology*, 105(1):160–164, jan 1998.
- [10] A. Kun and W. Miller. Adaptive dynamic balance of a biped robot using neural networks. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 1, pages 240–245. IEEE, 1996.
- [11] V. Lepetit, F. Moreno-Noguer, and P. Fua. EPnP: An Accurate O(n) Solution to the PnP Problem. *International Journal of Computer Vision*, 81(2):155–166, feb 2009.
- [12] M. Lourakis. A Brief Description of the Levenberg-Marquardt Algorithm Implemented by levmar. *Foundation of Research and Technology*, 4(1):1–6, 2005.
- [13] S. Martinez, J. M. Garcia-Haro, J. Victores, A. Jardon, and C. Balaguer. Experimental Robot Model Adjustments Based on Force-Torque Sensor Information. *Sensors*, 18(3):836, mar 2018.
- [14] G. G. Muscolo, C. T. Recchiuto, and R. Molino. Vision and locomotion control systems on a bio-inspired humanoid robot. In *MELECON 2014 - 2014 17th IEEE Mediterranean Electrotechnical Conference*, pages 380–385. IEEE, apr 2014.
- [15] A. Vakhitov, J. Funke, and F. Moreno-Noguer. Accurate and linear time pose estimation from points and lines. In *European Conference on Computer Vision*, pages 583–599. Springer, Cham, 2016.
- [16] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-

point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):930–943, aug 2003.



© 2018 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution CC-BY-NC 3.0 license (<http://creativecommons.org/licenses/by-nc/3.0/>).