

# Estrategia de prevención de colisiones en 3D para colaboración segura hombre-robot, usando el Kinect

## 3D collision prevention strategy for safe man-robot collaboration, using the Kinect

Alessandro Saffioti<sup>1\*</sup>, Felipe Pan<sup>2</sup>, Humberto Rodríguez<sup>1</sup>

<sup>1</sup>Ingeniería Mecánica, Facultad de Ingeniería Mecánica, <sup>2</sup>Ingeniería Aeronáutica, Facultad de Ingeniería Mecánica, <sup>1,2</sup>Universidad Tecnológica de Panamá.

**Resumen** Actualmente el uso de robots en la industria de manufactura requiere que el operador se separe del mismo mediante vallas o barreras físicas por temas de seguridad. El objetivo de este trabajo es desarrollar un sistema que le permita a las personas interactuar con un brazo robótico, en un mismo espacio de trabajo, de forma segura, al prevenir colisiones en tiempo real mediante la identificación y localización de la persona (operador o usuario) dentro del área de trabajo referenciado a partir del robot. Usualmente la segmentación de objetos en una imagen se hace a partir de diferencia de colores. Si se muestran múltiples objetos que poseen el mismo color, no se podrá segmentar la imagen utilizando solamente la información de este. Por este motivo, se introduce como información adicional, la profundidad, para descartar objetos que no están en el mismo plano de los objetos de interés. Esta es la razón por la cual se plantea la solución del problema con la ayuda de las imágenes de color y de profundidad proporcionadas por el sensor Kinect.

**Palabras clave** *Bounding box*, detección de colisiones, diferencias finitas, imagen de profundidad, imagen RGB, Kinect, MATLAB, matrices de transformación homogénea, parámetros de Denavit-Hartenberg, segmentación de imágenes.

**Abstract** Currently the use of robots in the manufacturing industry requires the operator to separate from it by means of fences or physical barriers for safety reasons. The objective of this work is to develop a system that allows people to interact with a robotic arm, in the same workspace, safely, by preventing collisions in real time by identifying and locating the person (operator or user), within the work area referenced from the robot. Usually the segmentation of objects in an image is done based on color difference. If multiple objects that have the same color are displayed, you cannot segment the image using only the information in the image. For this reason, depth is entered as additional information to discard objects that are not in the same plane as the objects of interest. This is the reason why the solution of the problem arises with the help of the color and depth images provided by the Kinect sensor.

**Keywords** *Bounding box*, collision detection, finite differences, depth image, RGB image, Kinect, MATLAB, homogeneous transformation matrices, Denavit-Hartenberg parameters, image segmentation.

---

\*Corresponding author: [alessandro.saffioti@utp.ac.pa](mailto:alessandro.saffioti@utp.ac.pa)

### 1. Introducción

La detección de colisiones es un tema de gran importancia en cualquier ambiente industrial que aplique la robótica. Los errores de programación, eventos inesperados u objetos mal posicionados pueden llevar a la colisión, si se basan en datos previamente provisionados del área de trabajo. El uso de sensores puede presentar una solución para que se puedan usar robots en un mismo espacio de trabajo con personas, pero esta solución presenta numerosos desafíos. La obstrucción del área

de trabajo por equipo, personas y demás puede interferir con los sensores para que perciban completamente el espacio.

Actualmente, el uso del robot en la industria requiere que la persona se separe por completo del mismo por vallas o barreras físicas. Ya que en la industria se utilizan robots de grandes dimensiones y de rápido movimiento, estos pueden provocar lesiones corporales graves e incluso la muerte; que limitan los movimientos del operador cuando trabajan en conjunto. La separación entre robots y operadores garantiza la seguridad, pero esta práctica es ineficiente por diferentes razones. El área de trabajo del robot puede ocupar mucho

espacio, inclusive si solo se utiliza una pequeña porción de esta. Por lo general, la operación del robot debe detenerse para que el operador pueda entrar en la zona de trabajo, ya sea para dejar o retirar material, lo cual se traduce en pérdidas de dinero, espacio y tiempo de producción [1].

Para solucionar el problema, proponemos el uso del sensor Kinect y un programa, cuya versión de prueba está desarrollada en MATLAB, para calcular en tiempo real si hay alguna interferencia entre el operador y, en este caso, al robot Nachi MZ04.

## 2. Antecedentes

La interacción segura entre humanos y robots ha sido extensamente estudiada desde el principio de la robótica. Los métodos existentes pueden ser clasificados en enfoques pre-colisión y post colisión. El método del enfoque pre colisión intenta prevenir la colisión, detectándolo de ante mano. Este método incluye sensores de proximidad montados en el robot o en el entorno. El método del enfoque post colisión detecta una colisión cuando ocurre y trata de reducir los daños resultantes. Los métodos en esta categoría detectan las colisiones a partir de sensores de fuerza y torque en el robot, limitando las fuerzas y la velocidad de las articulaciones del robot. Ninguno de los métodos post colisión previene la colisión, lo que los hace ineficientes en los sistemas de seguridad [1].

Independientemente la aplicación de la seguridad en robots, el problema de detectar y rastrear personas ha sido estudiado extensamente en el campo de visión [1]. Los métodos de visión usando cámaras funcionan bastante bien cuando las personas están bien separadas, sin obstrucciones y en posiciones neutras. Los sensores 3D pueden detectar personas en poses arbitrarias y el hardware del Kinect ha probado ser bastante confiable para la estimación de la posición de las personas [2], [3].

Muchos de los métodos anteriormente presentados aún no han sido adoptados por las industrias. La industria robótica logra, generalmente, un entorno seguro a través de separaciones como barreras físicas.

## 3. Formulación del problema

Si se dispone previamente de la trayectoria que seguirá el robot y la coordenada final del actuador, se puede determinar las posiciones de las articulaciones restantes a través de los parámetros de Denavit-Hartenberg (D-H) y la cinemática inversa.

Para lograr un entorno seguro y que los operadores puedan interactuar con los robots, se debe monitorear constantemente la zona de trabajo. Esto se logra con el Kinect, el cual con un arreglo de haces infrarrojos detecta los cuerpos dentro del área

de interés. A partir de los datos captados por el Kinect, se procede a segmentar la imagen e identificar entre los cuerpos, cual pertenece al robot y al operador. Con la información obtenida se crea un *bounding box* alrededor de los brazos del operador.

La forma propuesta para detectar si existe intercepción entre el robot y el operador es calcular la distancia más pequeña entre los *bounding boxes*: si esta distancia es menor a un valor establecido bajo los criterios propuestos en la investigación, se manda una señal al robot para que detenga o reduzca la velocidad de ejecución de los movimientos del robot y así evitar la colisión.

## 4. Kinect y conexión con MATLAB

El Kinect figura 1, es un sensor de movimiento que fue primeramente diseñado para los videojuegos, pero al que se le han encontrado muchas otras aplicaciones. Está compuesto por un sensor de profundidad 3D, una cámara RGB y un arreglo de cuatros micrófonos. El Kinect proporciona, en tiempo real, información RGB-profundidad a 30Hz con una resolución de 640 x 480 pixeles. Posee una vista angular de 57° horizontalmente y 43° verticalmente. El rango de profundidad oscila entre los 80 cm y 400 cm [2].



Figura 1. Kinect.

La información proporcionada por el Kinect puede ser adquirida en una computadora al descargar la versión adecuada de los drivers del Kinect (KinectSDK y KinectDeveloperToolkit). Adicionalmente, la comunicación del Kinect con la computadora se dificulta si se tiene alguno de los Kinect para Xbox, ya que estos no están diseñados con este propósito. Se necesita un cable que hace de fuente de alimentación y entrega la diferencia de voltaje necesaria para operar correctamente [4].

Por el motivo de que se ha utilizado ampliamente el Kinect en aplicaciones de visión y robótica, se le ha dado soporte en MATLAB (desde la versión 2013<sup>a</sup> en adelante) con el *Image Acquisition Toolbox Support Package for Kinect*.

## 5. Metodología

### 5.1 Cinemática inversa

La cinemática inversa resuelve la configuración que debe adoptar el robot para una posición y orientación del extremo

conocidas. El método para encontrar los ángulos de articulación requeridos para colocar la matriz de transformación homogénea  $\{T\}$  relativa al robot  $\{B\}$  se divide en dos partes [5]:

- Obtener las matrices de transformación homogéneas que relacionan la posición de la muñeca  $\{S_6\}$  con respecto a la posición de la base  $\{S_0\}$ .
- Realizar el procedimiento de cinemática inversa para resolver los ángulos de articulación.

En robótica la forma para describir la relación que existe entre dos elementos contiguos se realiza con los parámetros de D-H.

Así, asociando a la base del robot un sistema de referencia fijo (OXYZ) y al extremo un sistema de referencia que se mueva con él, se localizan los sistemas de referencias de cada una de las articulaciones del robot figura 2. Posteriormente, se determinan los parámetros D-H, con los que se construye la tabla 1.

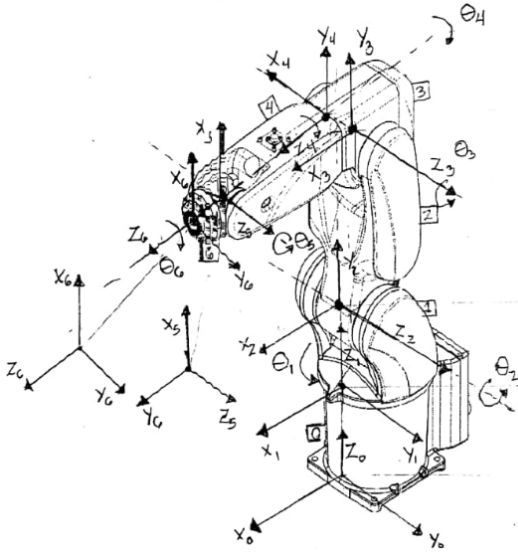


Figura 2. Ejes de referencia para el robot Nachi MZ04.

Tabla 1. Parámetros de D-H del robot. Fuente: Elaboración propia

Articulación	$\theta_i$	$d_i$	$a_i$	$\alpha_i$
1	q1	d1	0	0
2	q2	d2	0	90
3	q3	0	a3	0
4	q4	0	0	90
5	q5	d5	0	90
6	q6	d6	0	0

A cada eslabón se le puede asociar un sistema de referencia solidario a él y, utilizando las transformadas homogéneas, es posible representar las rotaciones y traslaciones relativas entre los distintos eslabones que componen al robot. Así se tiene que

la posición y orientación del eslabón final vendrá dada por la matriz T.

$$T = {}^0B {}^1B {}^2B {}^3B {}^4B {}^5B {}^6B = {}^0B \quad (1)$$

$$T = \begin{bmatrix} (ab) + (cd) & (gb) + (hd) & 0 & (ib) + (jd) \\ (ae) + (cf) & (ge) + (hf) & 0 & (ie) + (jd) \\ 0 & 0 & 1 & k \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Donde:

$$\begin{aligned} a &= (C_1C_2 - S_1S_2)(C_3C_4 - S_3S_4) + (-C_1S_2 - S_1C_2)(S_5C_6 + C_5S_6) \\ b &= (C_5C_6 - S_5S_6) \\ c &= (S_1C_2 + C_1S_2)(C_3C_4 - S_3S_4) + (-S_1S_2 + C_1C_2)(S_3C_4 + C_3S_4) \\ d &= (-C_5S_6 - S_5C_6) \\ e &= (S_5C_6 + C_5S_6) \\ f &= (-S_5S_6 + C_5C_6) \\ g &= (C_1C_2 - S_1S_2)(-C_3S_4 - S_3C_4) + (C_1S_2 - S_1C_2)(-S_3S_4 + C_3C_4) \\ h &= (S_1C_2 - S_1S_2)(-C_3S_4 - S_3C_4) + (C_1S_2 - S_1C_2)(-S_3S_4 + C_3C_4) \\ i &= (C_1C_2 - S_1S_2)(-S_3d_3) + (-C_1S_2 - S_1C_2)(C_3d_3) \\ j &= (S_1C_2 + C_1S_2)(-S_3d_3) + (-S_1S_2 + C_1C_2)(C_3d_3) \\ k &= (d_1 + d_2) + (d_4) \end{aligned} \quad (3)$$

Si se tiene la relación  $r_{xyz} = T r_{uvw}$  y se multiplica en ambos miembros por  $T^{-1}$ , se tiene:  $T^{-1} r_{xyz} = r_{uvw}$  por lo que se deduce que las vectores filas de la sub-matriz de rotación de la matriz T (vectores columna de la sub-matriz de rotación de  $T^{-1}$ ), representan los ejes principales del sistema de coordenadas de referencia OXYZ con respecto a O'UVW [1].

$$[{}^0B]^{-1} T = {}^1B {}^2B {}^3B {}^4B {}^5B {}^6B = {}^1B \quad (4)$$

De las 12 relaciones establecidas en la ecuación cuatro (4) interesan aquellas que expresan q1 en función de constantes para luego poder calcular los demás ángulos.

### 5.2 Resolución de sistemas de matrices homogéneas

Las coordenadas homogéneas son la representación de la posición en el espacio de un sólido con (n+1) dimensiones, de tal forma que un vector p (x, y, z) vendrá representado por p (w x, w y, w z, w), donde w tiene un valor arbitrario y representa un factor de escala; si en cambio, w es igual a 0 es una dirección [6].

$$p = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} aw \\ bw \\ cw \\ w \end{bmatrix} = \begin{bmatrix} a \\ b \\ c \\ 1 \end{bmatrix} \quad (5)$$

A partir de la definición de las coordenadas homogéneas, surge inmediatamente el concepto de matriz de transformación homogénea. Se define como matriz de transformación homogénea T a una matriz de dimensión 4x4 que representa la transformación de un vector de coordenadas homogéneas de un sistema de coordenadas a otro [5].

$$T = \begin{bmatrix} R_{3 \times 3} & p_{3 \times 1} \\ f_{1 \times 3} & w_{1 \times 1} \end{bmatrix} = \begin{bmatrix} Rotacion & Traslacion \\ Perspectiva & Escalado \end{bmatrix} \quad (6)$$

Se puede considerar que una matriz homogénea se haya compuesta por cuatro sub-matrices de distinto tamaño: una sub-matriz  $R_{3 \times 3}$  que corresponde a una matriz de rotación, una sub-matriz  $p_{3 \times 1}$  que corresponde al vector de traslación, una sub-matriz  $f_{1 \times 3}$  que representa una transformación de perspectiva y una sub-matriz  $w_{1 \times 1}$  que representa un escalado global [5].

A partir del Kinect {A} se sacan dos matrices de transformación homogénea: una hacia el robot {B} y la otra hacia el operador {C} ver figura 3.

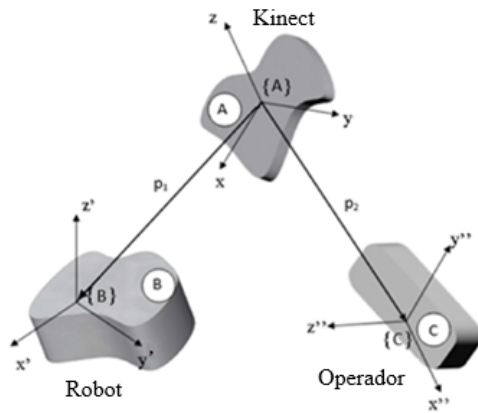


Figura 3. Conjunto de transformadas a partir del punto A. Fuente: Elaboración propia.

$${}^A_B T = Rotz(\theta)T(p_1) = \begin{bmatrix} C\theta & -S\theta & 0 & x_1 \\ S\theta & C\theta & 0 & y_1 \\ 0 & 0 & 0 & z_1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad (7)$$

$${}^A_C T = Rotz(\phi)T(p_2) = \begin{bmatrix} C\phi & -S\phi & 0 & x_2 \\ S\phi & C\phi & 0 & y_2 \\ 0 & 0 & 0 & z_2 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad (8)$$

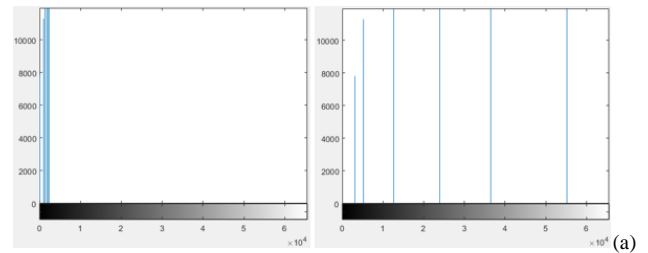
Donde la matriz de transformación homogénea hacia el robot ( ${}^A_B T$ ) se debe realizar para cada articulación y  $p_1 \{x_1, y_1, z_1\}$  representa las distancias entre el Kinect y robot,  $p_2 \{x_2, y_2, z_2\}$  representa las distancias entre el Kinect y el operador,  $\theta$  representa la rotación del robot y  $\phi$  representa la rotación del operador.

### 5.3 Segmentación de la imagen

Se logra la segmentación de imágenes con la ayuda de la información de profundidad proporcionada por el Kinect. Primeramente, se hace una ecualización en el histograma, para así corregir el bajo rango dinámico de la imagen [5].

Esto se puede apreciar en la figura 4 (a), en donde el histograma se concentra en la parte oscura de la escala de

grises. La imagen en la figura 4 (b) es el resultado de la ecualización del histograma, donde las mejoras en la intensidad promedio y contraste son bastante evidentes.



(b)

Figura 4. Ilustración de los histogramas, (a) Histograma de la imagen original, (b) Ecualización del histograma.

Una vez obtenida la imagen con el histograma ecualizado se descartan todos los valores que sobrepasan una cierta distancia donde no ocurre nada de interés figura 5. Con esta técnica se elimina el fondo y se segmenta con gran facilidad la imagen.

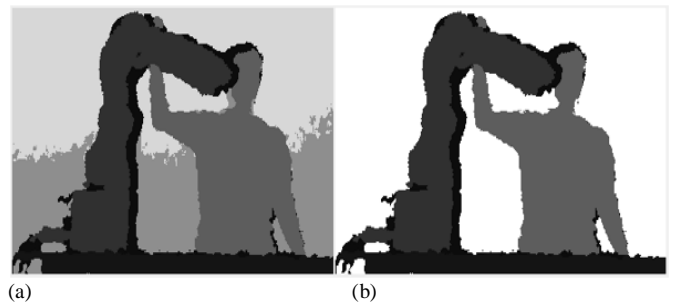


Figura 5. (a) Imagen de profundidad con la ecualización del histograma, (b) Segmentación de la imagen.

### 5.4 Identificación de la persona y el robot

La identificación de la persona y el robot se obtiene al transformar la imagen de color a una escala de grises y calcular la media de la imagen segmentada en la imagen de color figura 6. La lógica detrás de esta técnica es que las intensidades en los píxeles del robot son, en su mayoría, idénticos por lo que las medias de estos valores serán mayores que aquellos presentados por los píxeles que representen a la persona.

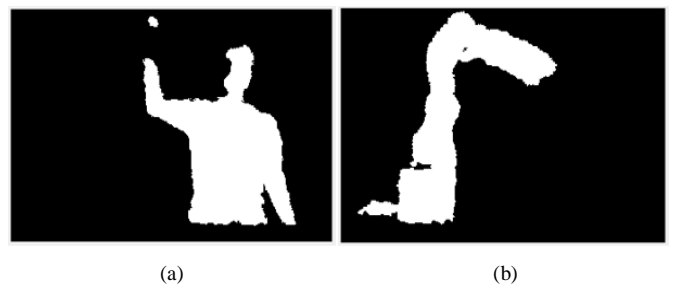


Figura 6. (a) Identificación de la persona (b) Identificación del robot.

### 5.5 Detección de colisiones

Para asegurarse que cualquier área en el espacio no pueda estar ocupada por más de un objeto, se necesita la detección de colisiones basada en la información geométrica del objeto.

El inconveniente con la detección de colisiones es que se debe probar con cada punto del objeto, lo cual resulta en un gran tiempo computacional y mucha capacidad de procesamiento. Una forma para solucionar este inconveniente es dividir el espacio, por ejemplo, en cubos y así solo se debe probar la colisión para los vértices de los cubos. Estos cubos son llamados *bounding boxes*.

En 2D si los vértices de los *bounding boxes* se superponen hay colisión, para corroborar tal enunciado en 3D se debe verificar la información de profundidad. Por lo que se crea un *bounding box* en la persona y en el robot figura 7. Para luego calcular la distancia mínima que existe entre los objetos de interés [6].

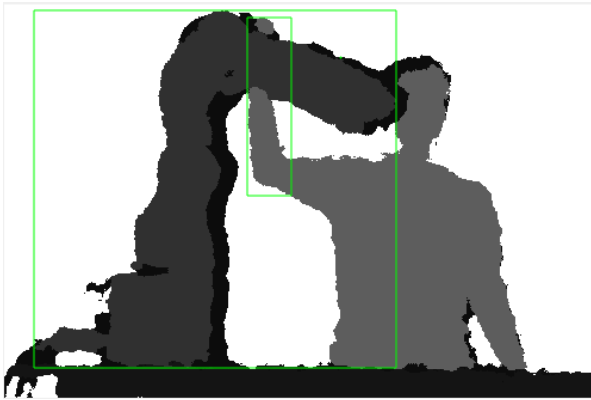


Figura 7. *Bounding box* alrededor del brazo de la persona y el robot.

Si los vértices de los *bounding boxes* de los dos objetos se representan por  $\{(x_1, y_1, z_1), \dots, (x_m, y_m, z_m)\}$  y  $\{(r_1, s_1, t_1), \dots, (r_m, s_m, t_m)\}$ , la distancia mínima entre los objetos se representa por:

$$d = \min(\sqrt{(x_i - r_j)^2 + (y_i - s_j)^2 + (z_i - t_j)^2}) \quad (9)$$

La ecuación 9, es la distancia euclidiana; que, en geometrías euclidianas, representa la distancia más pequeña que existe entre dos puntos, en este caso, a través de un espacio en 3D.

Para facilitar el trabajo se hace la suposición de que el torso y la cabeza de la persona se pueden despreciar quedándose con los brazos.

### 5.6 Estimación de la posición y velocidad

En busca de mejorar los resultados anteriores se decide implementar un algoritmo para estimar la posición y velocidad del brazo de la persona. Esta práctica logra optimizar el código anterior en situaciones específicas como, por ejemplo: Si la

distancia entre el robot y el brazo de la persona es pequeña pero este último no se mueve, no es necesario detener el robot.

La derivada de una función tiene muchas aplicaciones, entre las cuales esta la determinación de la velocidad instantánea de una partícula o móvil a partir de su función de posición, por lo que, se emplea las diferencias finitas hacia atrás para calcular la velocidad del brazo de la persona en todo momento, utilizando la posición del *bounding box* como referencia.

$$f'(x) = \frac{x_{i+1} - x_i}{\Delta t} + \frac{y_{i+1} - y_i}{\Delta t} + \frac{z_{i+1} - z_i}{\Delta t} \quad (10)$$

## 6. Resultados

### 6.1 Distancia obtenida

La distancia mínima calculada entre los *bounding boxes* de la persona y el robot, a través del programa escrito en MATLAB, es de 32.3 cm.

### 6.2 Tiempo de procesamiento

El tiempo de procesamiento con el tipo de resolución de imagen se muestra en la tabla 2.

Tabla 2. Tiempo de procesamiento para la resolución estándar del Kinect

Resolución [pixel x pixel]	FPS [Hz]	Tiempo [segundos]
640 x 480	30	0.4811

El código se probó en el siguiente sistema:

- Software: MATLAB 2017<sup>a</sup>
- OS: Microsoft Windows 10, 64 bits
- Hardware: NVIDIA GeForce GTX 650
- Procesador: Intel® Core(TM) i7-3770K CPU @ 3.50GHz (8 CPUs), ~3.5GHz

## 7. Conclusiones

Este documento presenta un sensor en tiempo real de un sistema que está diseñado para la seguridad de las personas que operan en zonas cercanas al área de trabajo de los robots en procesos industriales. Este está basado en el método de "*bounding box*" para la detección de objetos, cuyo objetivo fue trabajar con los vértices del área acotado por el cubo y no con la geometría propia del objeto analizado, con el propósito de ahorrar capacidad y tiempo de procesamiento computacional.

Con este método se calcula que la distancia entre los objetos es de 32.3 cm. También se calculó que el tiempo de procesamiento fue de 0.4811 segundos para realizar toda la operación.

Para calcular la distancia mínima entre distintos puntos de forma eficiente utilizando la ecuación euclidiana, los objetos



de interés deben estar separados por una distancia considerable entre ellos. Pero cuando esta distancia es pequeña y los vértices de los *bounding boxes* no representan la figura del objeto adecuadamente habrá un error considerable entre la distancia real y la distancia calculada. Esto se observa en la figura 7, donde el *bounding box* del robot ocupa mucho espacio. A medida que la distancia entre los objetos de interés se acorta, el error de medición se aumenta y, a manera inversa, mientras que la distancia aumenta, el error disminuye.

En el futuro se planteará la adquisición de datos por varios Kinect ya que, se podría presentar problemas al separar los objetos de interés si existe oclusión entre ellos. Además, se presentará en modo de comparación el uso de las redes neuronales para agilizar este procedimiento como también otras técnicas de detección de colisiones con mejores resultados, como lo es el filtro Kalman.

## AGRADECIMIENTOS

Al profesor Humberto Rodríguez, jefe del Laboratorio Especializado en Análisis, Diseño y Simulación (LEADS), se le agradece por todo el apoyo brindado en la realización de este trabajo. Al igual que a todos los compañeros que ayudaron en

la elaboración del documento de una forma u otra, en especial a Javier Alexander Bernal.

## REFERENCIAS

- [1] P. Rybski, P. Anderson-Sprecher, D. Huber, C. Niessl, R. Simmons. "Sensor Fusion for Human Safety in Industrial Workcells".
- [2] A. Abramov, K. Pauwels, J. Papon, F. Worgoteer & B. Dellen. "Depth-supported real time video segmentation with Kinect".
- [3] J. Hernández, A. Quintanilla, J. López, F. Rangel, M. Ibarra & D. Almanza. "Detecting objects using color and depth segmentation with Kinect sensor".
- [4] S. E. Bernal Saira. "Sistema para el análisis cinemático de la marcha humana usando técnicas de procesamiento de imágenes". Tesis de pregrado, Universidad Tecnológica de Panamá, 2016.
- [5] John J. Craig. "Robótica", tercera edición, Pearson Education.
- [6] A. Barriento, L. F. Peñin, C. Balaguer, R. Aracil. "Fundamentos de Robótica", segunda edición, Distribuidora Mc GrawHill.
- [7] R. Gonzalez, R. Woods & S. Eddins. Digital Image Processing using MATLAB. Pearson Prentice Hall, vol. 2, pp. 81-83.
- [8] F. Gong, B. Gao & Q. Niu. "An Algorithm for Rapidly Computing the Minimum Distance between two Objects Collision Detection".