

Evasión de obstáculos en tiempo real usando visión estéreo

Taihú Pire

Departamento de Computación, Facultad de Ciencias Exactas y Naturales, UBA
tpire@dc.uba.ar

Resumen— En este trabajo se presenta un método de evasión de obstáculos para la navegación autónoma de robots móviles usando una cámara estéreo como único sensor. A partir de las imágenes izquierda y derecha, se calcula la disparidad de cada punto en el mundo. De esta forma podemos estimar la distancia a los objetos y definir una heurística para guiar al robot hacia la zona libre de obstáculos. El método propuesto ha sido probado exitosamente en ambientes interiores. Además se detalla la implementación de los algoritmos involucrados que alcanzan una ejecución en tiempo real.

I. INTRODUCCIÓN

La tarea de evasión de obstáculos es uno de los problemas más importantes y de mayor interés en el campo de la robótica móvil. Para poder navegar en el mundo real, es necesario detectar aquellas zonas del mundo que son peligrosas o difíciles de atravesar. Para una gran cantidad de problemas de navegación el mundo puede ser modelado como un plano, de esta manera cualquier punto que no pertenezca al plano puede ser considerado un obstáculo. La mayoría de los algoritmos de evasión de obstáculos utilizan sensores de distancia, tales como sonares [1], láser [2] y radares [3]. Estos sensores son adecuados para la tarea de detección de obstáculos y pueden ser utilizados para distintos propósitos, dado que pueden medir las distancias de los obstáculos al robot. Sin embargo, ninguno de estos sensores puede llevar a cabo perfectamente la tarea de detección de obstáculos. Los sonares tienen un bajo costo pero sufren de reflexión especular y usualmente cuentan con un ángulo de resolución muy acotado. Los láser y radares proveen una mejor resolución, pero son más complejos y costosos. Los sensores visuales son una solución alternativa para la evasión de obstáculos, y su popularidad se ha incrementado significativamente en el campo de la robótica móvil [4] [5]. Estos sensores frecuentemente proveen una mejor resolución de los datos y rangos de medición más grandes. Además, dado que los sensores visuales son pasivos, son menos dependientes del entorno.

Para llevar a cabo la evasión de obstáculos, un robot necesita conocer la distancia de los objetos de su alrededor. El método más común para la extracción de información de profundidad de imágenes visuales es la visión estéreo. La visión estéreo frecuentemente produce mapas de profundidad precisos, sin embargo la desventaja de esta tecnología es que requiere una potente plataforma de procesamiento.

En el presente informe se exhibe un método de evasión de obstáculos que usa una cámara estéreo como único sensor para la navegación autónoma de un robot. Dicho

método utiliza *mapas de disparidad* que permiten aproximar las distancias en las que se encuentran los objetos del entorno con respecto al robot. La implementación del método propuesto alcanza una ejecución en tiempo real utilizando como plataforma de procesamiento una notebook estándar.

El documento se encuentra estructurado de la siguiente manera: la sección II introduce el estado del arte en sistemas de evasión de obstáculos mediante el uso de visión estéreo, la sección III describe el algoritmo de evasión de obstáculos propuesto, la sección IV presenta algunos aspectos de implementación y las librerías utilizadas para el procesamiento de imágenes y la generación de mapas de disparidad, la sección V presenta los resultados de los experimentos realizados, finalmente la sección VI exhibe las conclusiones y posibles trabajos futuros a desarrollar.

II. TRABAJO RELACIONADO

Existen varios trabajos que utilizan visión estéreo para la evasión de obstáculos de robots móviles. En [6] se comparan tres métodos para llevar a cabo la tarea de evasión de obstáculos en base a la utilización de mapas de disparidad. El primer paso de los métodos es común a los tres: se propone dividir el mapa de disparidad en tres partes: una izquierda, una central y una derecha. La diferencia entre estos métodos reside en cómo se procesa la información a partir de aquí. En el primer método se propone calcular el promedio de disparidad en cada una de las zonas. Entonces, se dirige al robot en la dirección correspondiente a la zona con el menor de los valores obtenidos, asumiendo que es aquella asociada a la menor cantidad de obstáculos. El segundo método consiste en una mejora del primero. En principio sólo se considera la parte central del mapa de disparidad. Se realiza un conteo sobre los píxeles que presentan una disparidad mayor a un cierto umbral, y se calcula el porcentaje que estos representan en relación al total de esa zona particular. Si este porcentaje es alto, se considera que frente a la cámara se encuentra un obstáculo. Es entonces que se decide realizar una acción evasiva. Para poder elegir una dirección hacia la cual dirigir al robot, se analizan las zonas laterales y se procede hacia la que contenga el menor promedio de disparidad. El último método considera la condición de que las tres zonas contengan un promedio de disparidad por encima de un umbral preestablecido. Si el robot se encuentra en este caso, deberá actuar con una acción de escape, por ejemplo, girar 180 grados, o volver en reversa sobre el camino por donde se llegó a la posición actual.

En [7] se sugiere el uso de sólo una parte de la imagen, un área pequeña en comparación al tamaño total de la misma. Esto es lo que se conoce como *región de interés*. De esta forma, se logra reducir el tiempo requerido para poder obtener el resultado. Un problema que puede presentar este método es que, si el área de interés es demasiado pequeña, se ignora información relevante sobre la escena a procesar.

En el trabajo actual se desarrolla un nuevo método de evasión de obstáculos donde se toman como punto de partida los trabajos [6] y [7], y se los extiende para lograr un mejor comportamiento, alcanzando una ejecución en tiempo real.

III. DESCRIPCIÓN DEL MÉTODO

En esta sección se describe el algoritmo de evasión de obstáculos, el mismo consiste de tres etapas. En la primera etapa se realiza el procesamiento previo de las imágenes, necesario para el cálculo de mapas de disparidad consistente en la rectificación y alineación vertical de las mismas. La segunda etapa consiste en el cálculo de los mapas de disparidad a partir de las imágenes ya procesadas. La tercera etapa es el algoritmo de control basado en la heurística de evasión, es decir, la forma en que es analizado un mapa de disparidad en cada instante de tiempo y el comportamiento que realiza el robot.

A. Visión estéreo

La *geometría epipolar* estudia las relaciones geométricas entre los puntos del mundo (3D) y las proyecciones de los mismos en las imágenes (2D) capturadas por dos cámaras [8]. La Fig. 1 muestra las relaciones entre el punto \mathbf{x} de la imagen izquierda, el punto \mathbf{X} en el mundo real y la línea epipolar l' de la imagen derecha. La Fig. 2 ilustra las proyecciones, \mathbf{x} y \mathbf{x}' , de un punto \mathbf{X} en el espacio tridimensional en los planos de las imágenes tomadas por las cámaras con centros \mathbf{C} y \mathbf{C}' respectivamente. Al plano definido por la línea base y el punto \mathbf{X} del mundo se lo denomina *plano epipolar* π . Haciendo uso de estas relaciones se puede determinar la posición de un punto en el mundo dado un par de imágenes.

Para el método de evasión de obstáculos aquí presentado, se hizo uso de *mapas de disparidad*, que contienen información de profundidad de cada punto del mundo con respecto a la cámara. Una condición necesaria antes de poder realizar el cálculo de mapas de disparidad, consiste en tener ambas imágenes de manera tal, que los píxeles que se correspondan entre una imagen y otra se encuentren a la misma altura, es decir, que las líneas epipolares estén en las mismas filas de las imágenes. Por esta razón debe realizarse una rectificación de las imágenes y una alineación vertical (de ambas) de forma que, si se traza una línea horizontal, esta pase sobre el par de puntos correspondientes (ver Fig. 3).

B. Mapas de disparidad

En la Fig. 4 se puede observar la geometría resultante luego del procesamiento de las imágenes. Una vez rectificadas y alineadas verticalmente las imágenes, se procede

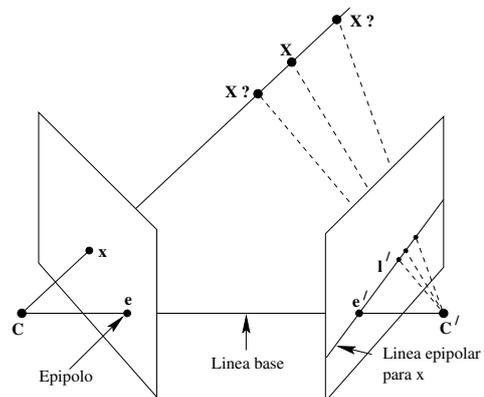


Fig. 1. Rayo definido por el centro de la cámara \mathbf{C} y la retro-proyección del punto \mathbf{x} de la imagen izquierda. La línea epipolar l' es la imagen de dicho rayo. Las proyecciones de los puntos \mathbf{X} proyectados por la cámara \mathbf{C}' se ubican sobre l' . Imagen adaptada de [8].

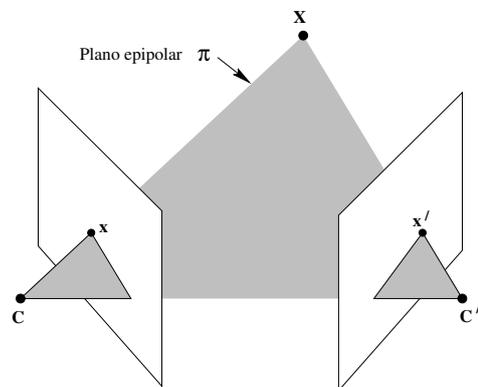
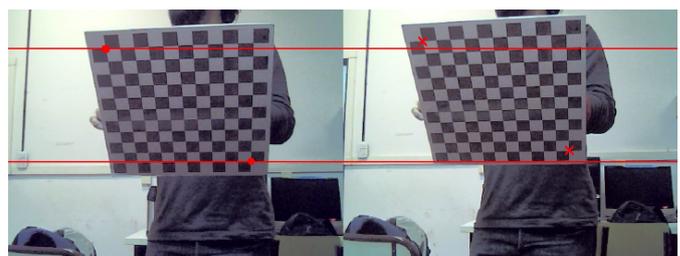
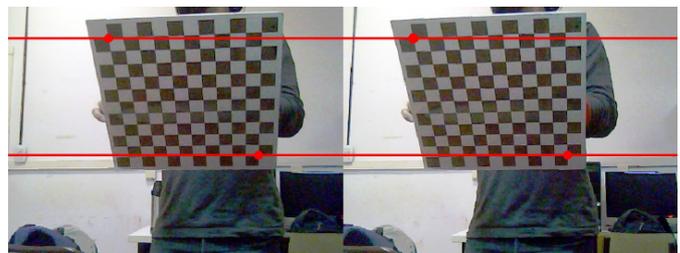


Fig. 2. Geometría epipolar. Imagen adaptada de [8].



(a) Capturas durante la calibración de la cámara.



(b) Imágenes luego de la rectificación y alineación vertical.

Fig. 3. Imágenes donde se ilustra el resultado de la rectificación.

al cálculo del mapa de disparidad. Se llama *disparidad* a la distancia de un píxel en una imagen con su correspondiente en la otra, asimismo, se define *mapa de disparidad* a la matriz constituida por estos valores. Aquellos píxeles pertenecientes a un objeto cercano, tendrán mayor disparidad que otros pertenecientes a un objeto lejano. Los puntos homólogos en cada imagen se buscan por medio de un algoritmo de correspondencia.

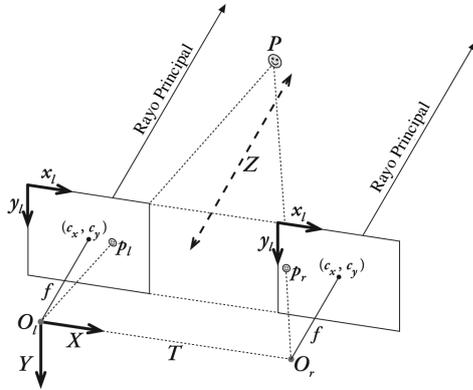


Fig. 4. Imágenes rectificadas y alineadas verticalmente. Imagen adaptada de [11].

Sean x^l y x^r las posiciones horizontales de los puntos correspondientes en la imagen izquierda y en la imagen derecha respectivamente, entonces la disparidad d está dada por $d = x^l - x^r$. Utilizando triángulos semejantes (1) es posible estimar la profundidad Z de un punto en el mundo dada la distancia focal f , la línea base T y su valor de disparidad d . En la Fig. 5 se ilustra el uso de triángulos semejantes geoméricamente.

$$\frac{T - (x^l - x^r)}{Z - f} = \frac{T}{Z} \rightarrow Z = \frac{fT}{x^l - x^r} \rightarrow Z = \frac{fT}{d} \quad (1)$$

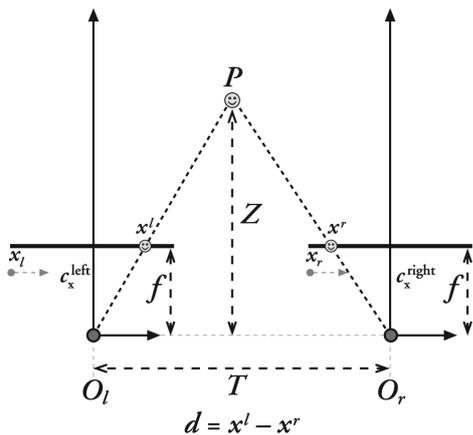


Fig. 5. La profundidad Z puede ser calculada haciendo uso de la propiedad de triángulos semejantes. Imagen adaptada de [11].

Observar que la distancia de un punto a las cámaras es inversamente proporcional al valor de disparidad (Fig. 6). Si el valor de disparidad es cercano a cero, pequeñas variaciones en la disparidad producen grandes variaciones en la profundidad, en cambio, si el valor de disparidad es

grande, pequeñas variaciones en la disparidad no producen diferencias de profundidad. Por lo tanto, si el valor de disparidad es muy pequeño o muy grande, la disparidad no brinda información sobre la distancia a la que se encuentra un objeto.

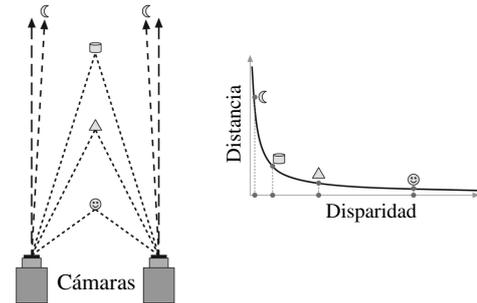


Fig. 6. La profundidad es inversamente proporcional a la disparidad, por lo tanto las mediciones son restringidas a objetos cercanos. Imagen adaptada de [11].

La Fig. 7 resume el procedimiento de obtención del mapa de disparidad dada una captura de una cámara estéreo. Luego de la rectificación y alineación vertical de las imágenes, se alinean horizontalmente una imagen sobre otra para definir cuál es la distancia más lejana que se desea considerar. Los puntos ubicados en este límite son denominados *puntos con disparidad cero* o *nula*. Dicho valor puede variar según el entorno. En general se toma como referencia el punto más lejano del ambiente en el cual se trabaja, por ejemplo, si el ambiente es reducido, es deseable que los puntos con disparidad cero se encuentren a una distancia más cercana que si el ambiente es muy grande, resultando en una alineación horizontal diferente para cada caso. Finalmente se procede a calcular el mapa de disparidad. Se representa el resultado en el espacio de colores HSV (ver Fig. 7(d)), de modo que se puede apreciar fácilmente la cercanía a los objetos.

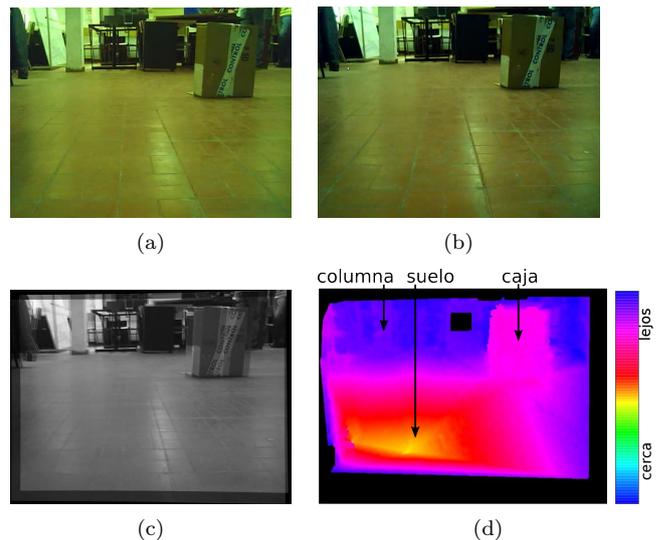


Fig. 7. Imágenes donde se muestran los resultados parciales durante el proceso de obtención del mapa de disparidad. (a) y (b) capturas originales de una cámara, izquierda y derecha respectivamente. (c) superposición de las imágenes izquierda y derecha, en escala de grises (d) mapa de disparidad resultante.

C. Evasi3n de obst3culos

El algoritmo de control para la evasi3n de obst3culos que se implementa est3 basado en el M3todo de Estimaci3n por Umbral (*Threshold Estimation Method*) analizado en [6]. Este m3todo utiliza mapas de disparidad para aproximar la distancia a la que se encuentran los objetos de la im3gen capturada por la c3mara est3reo. A continuaci3n, se presentan las variaciones implementadas sobre el algoritmo original.

En primer lugar, se divide el mapa de disparidad en tres ventanas verticales de tama3o configurable (l3neas rojas de la Fig. 8(a)). Se configur3 el ancho de la ventana central para que sea mayor que el de las laterales. De esta manera se intenta minimizar las limitaciones causadas por el 3ngulo de visi3n reducido de las c3maras, logrando as3 que la ventana central obtenga mayor informaci3n hacia sus costados. De este modo se intenta abarcar un 3rea suficiente para que se detecten los objetos que se encuentran delante como obst3culos potenciales. Adem3s, se configuraron las ventanas de los lados con un ancho distinto, a fin de reducir el ruido generado por la falta de informaci3n en los bordes de los mapas de disparidad (zonas negras en los bordes de la Fig. 8(b)). Por otro lado, se estableci3 un l3mite para determinar la regi3n de inter3s sobre la cual se procesan los datos como en [7] (l3nea verde de la Fig. 8(a)), prescindiendo as3 de la informaci3n proveniente del suelo, debido a que 3ste es filtrado como un “objeto” cercano. Es importante aclarar que el algoritmo asume que el terreno por donde se encuentra navegando el robot es plano.

El segundo paso consiste en estimar los porcentajes de p3xeles del mapa de disparidad que representen objetos cercanos, los cuales se obtienen para cada una de las ventanas por separado. Para esto, sobre cada 3rea se contabilizan los p3xeles que posean una disparidad mayor a un cierto umbral de tolerancia; mientras menor sea el umbral, mayor ser3 la cantidad de p3xeles considerados cercanos. Tambi3n, se tuvieron en cuenta aquellos p3xeles que no cuentan con informaci3n de disparidad (zonas negras de la Fig. 8(d)), dado que los mapas de disparidad en general presentan una gran cantidad de zonas en donde no se encuentran correspondencias entre p3xeles entre ambas im3genes. Para evitar que estos predominen sobre los p3xeles de alta disparidad, se los incorpora ponder3ndolos al 30% del valor de un p3xel con disparidad alta. Cabe destacar que, por motivos de eficiencia, las zonas de los costados no son procesadas hasta el momento en que realmente sea necesario, como veremos a continuaci3n.

Una vez obtenidos los porcentajes, se procede a tomar la decisi3n del pr3ximo movimiento. En un principio, se analiza 3nicamente la zona central. Si el porcentaje de p3xeles con disparidad alta es menor a un cierto valor predeterminado (por ejemplo, 30%), se considera que el camino se encuentra sin obst3culos cercanos y el robot puede continuar su desplazamiento en l3nea recta. En cambio, si el umbral es superado, se procede a comparar los porcentajes de p3xeles de alta disparidad de las ventanas laterales (cada uno respecto a la cantidad de p3xeles total de la ventana correspondiente). La ventana con menor valor asociado representa la direcci3n con menos

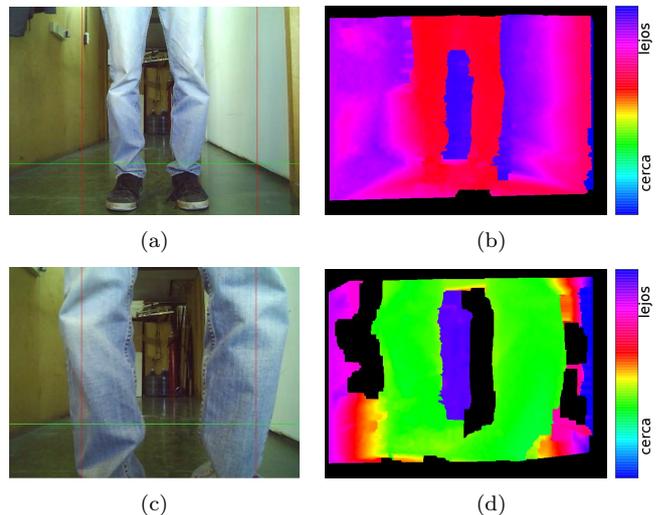


Fig. 8. Im3genes donde se muestran las ventanas divisorias junto con el l3mite que determina el 3rea de inter3s, y los correspondientes mapas de disparidad. (a) una persona a 1.5 metros de distancia a la c3mara y su respectivo mapa de disparidad en (b). (c) la persona se encuentra cercana a la c3mara (a 50 cent3metros aproximadamente), en (d) se puede observar que en la ventana central de la imagen predominan p3xeles con disparidad alta (p3xeles verdes).

obst3culos, y por lo tanto se procede a girar el robot sobre su eje en dicho sentido. Cabe aclarar que es posible configurar una velocidad angular del robot distinta a la velocidad lineal. Lo antedicho se puede utilizar para mitigar la informaci3n borrosa (efecto de blur en las im3genes) generada por el movimiento de rotaci3n, que conlleva la obtenci3n de mapas de disparidad corruptos.

IV. IMPLEMENTACI3N

El m3todo propuesto fue implementado en el lenguaje de programaci3n C++. Para la calibraci3n de las c3maras se utiliz3 una extensi3n de MATLAB llamada *Toolbox Calib* [9]. El resultado de esta calibraci3n est3reo es un archivo de texto plano con los par3metros intr3secos (relativos a la c3mara: distancia focal, punto principal, distorsi3n radial) y extr3secos (de la c3mara con respecto al mundo: rotaci3n y traslaci3n) de ambas c3maras. Con este archivo se genera un archivo XML, estructurado de forma tal, que puede ser le3do autom3ticamente por m3todos de la librer3a *OpenCV* [10] [11]. Esta 3ltima es utilizada para la rectificaci3n y alineaci3n vertical de las im3genes. Luego, para el c3lculo de mapas de disparidad se hizo uso de la librer3a LIBELAS (Library for Efficient LArge-scale Stereo Matching) [12]. Si bien esta librer3a tiene la capacidad de proveer dos mapas de disparidad por cada par de cuadros: el izquierdo con respecto a la derecha, y el derecho con respecto a la izquierda, para la implementaci3n del algoritmo aqu3 presentado s3lo se analiza el mapa izquierdo. La Tabla I muestra el tiempo (en milisegundos) que demoran las etapas del m3todo propuesto (rectificaci3n y alineaci3n, computo de mapas de disparidad y control de movimiento) durante el procesamiento de un cuadro. Se puede observar el alto costo del computo del mapa de disparidad.

Etapa del método	Tiempo en ms
Rectificación y alineación	~ 20
Mapa de disparidad	~ 200
Control de movimiento	~ 10

TABLA I

TIEMPO COMPUTACIONAL (EN MILISEGUNDOS) QUE TOMAN LAS ETAPAS DEL MÉTODO DURANTE EL PROCESAMIENTO DE UN CUADRO.

V. RESULTADOS

Para realizar las experiencias se utilizó el robot *Exabot* [13], diseñado y fabricado en la facultad *FCEyN-UBA*, Argentina (ver Fig. 9). Se trata de un vehículo móvil diferencial con cuatro ruedas, unidas por dos orugas, y dos motores, lo que permite movimientos con dos grados de libertad en el plano. Se configuró con una cámara estéreo y un montaje para computadoras portátiles. La interfaz de comunicación con el mismo es vía ethernet utilizando el protocolo UDP/IP, y se usa únicamente para enviar comandos a los motores.



Fig. 9. Configuración de Exabot utilizada.

El equipo de visión estéreo montado en el robot es una cámara *Minoru*. La misma se conecta vía USB a la computadora portátil. Sus características son las siguientes: 60 mm de base estéreo, USB 2.0, resoluciones desde 320×240 a 1280×480 , 15 \sim 30 cuadros por segundo, 1.5 W en modo operativo y 2 mW en modo suspendido. La resolución utilizada en este trabajo para las imágenes es de 640×480 , capturadas a 15 cuadros por segundo. Cabe aclarar que si bien la cantidad de imágenes disponibles por segundo es la mencionada anteriormente, las que realmente se procesan en general son menos, debido al costo computacional del método propuesto.

Las pruebas del método fueron llevadas a cabo sobre una computadora portátil con procesador Intel Core i5 \sim 2.67GHz Dual core, 4GB de RAM DDR3. Se logró alcanzar un procesamiento de 8 \sim 10 cuadros por segundo, suficientes para que el robot reaccione en tiempo real a una velocidad lineal y angular de 3 cm/s .

Los experimentos se realizaron en ambientes interiores, en habitaciones con no más de 10 m de distancia máxima

y en pasillos de longitud de más de 50 m . Además de las paredes propias del entorno, se ubicaron objetos de tamaños mayores a 40 cm de altura (altura de la cámara), y 30 cm de ancho, y se pudo observar un buen desempeño, esquivando todos los obstáculos correctamente. La Fig. 10 ilustra la trayectoria trazada por el robot durante la evasión de obstáculos en una habitación. La Fig. 11 muestra imágenes del robot durante la evasión de obstáculos en un pasillo. En [14] se puede ver el video del experimento.

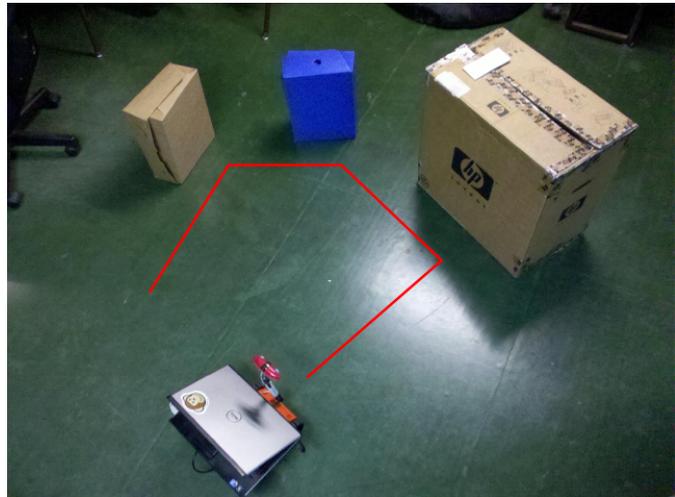


Fig. 10. Trayectoria realizada por el robot durante la evasión de obstáculos.

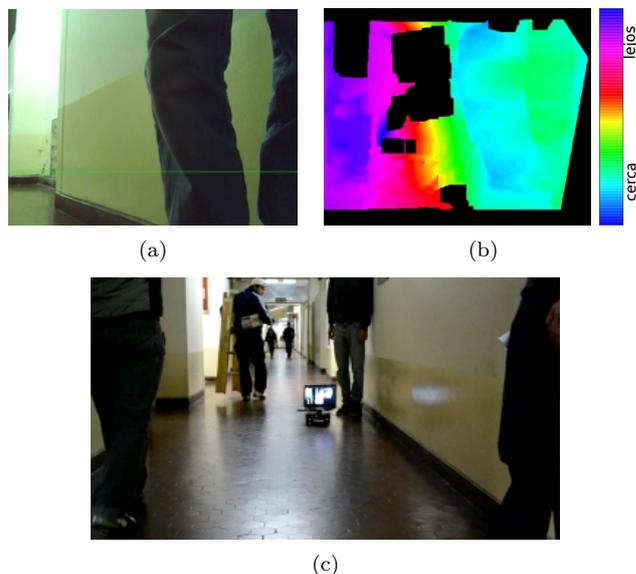


Fig. 11. Experimento en un pasillo: (a) captura de la cámara izquierda, (b) su correspondiente mapa de disparidad. (c) vista externa del robot.

Se pudo ver que la calidad de detección de los objetos depende directamente de su tamaño, textura y luminosidad. Esto se debe a que la búsqueda de correspondencia de píxeles entre las dos imágenes es imprecisa cuando el objeto tiene, por ejemplo una textura uniforme, como es el caso de una pared lisa. A su vez, también depende de la cámara utilizada, dado que si ésta posee una baja

definición, es probable que el desempeño del algoritmo no sea el esperado.

VI. CONCLUSIONES Y TRABAJO FUTURO

En este trabajo se presenta un método para la evasión de obstáculos basado en mapas de disparidad utilizando una cámara estéreo como único sensor. Durante la parte experimental, se observó el correcto funcionamiento del robot, esquivando todos los obstáculos del entorno en tiempo real. Por otro lado, se pudo verificar que la generación de los mapas presenta un alto costo computacional. Una posible optimización es calcular únicamente el mapa de disparidad correspondiente a cada sub-ventana de la imagen bajo demanda, es decir, calcular los mapas laterales sólo cuando no sea posible avanzar hacia adelante. También se espera complementar el presente trabajo calculando la retroproyección de puntos característicos de las imágenes para obtener una estimación de sus posiciones en el mundo real, y así obtener una nube de puntos 3D del entorno. En particular, dicha nube de puntos puede ser utilizada para la obtención de *mapas de altura* (considerando únicamente la componente y), y usarlos para detectar de forma precisa el suelo, permitiendo realizar una mejor heurística de evasión.

REFERENCIAS

- [1] Johann Borenstein and Yoram Koren. Obstacle avoidance with ultrasonic sensors. *IEEE Trans. Robot. Automat.*, page 1988, 1998.
- [2] Hartmut Surmann, Kai Lingemann, Andreas Nüchter, and Joachim Hertzberg. A 3d laser range finder for autonomous mobile robots. *in Proceedings of the 32nd International Symposium on Robotics (ISR '01)*, pages 153–58, 2001.
- [3] K. Kaliyaperumal, S. Lakshmanan, and K. Kluge. An algorithm for detecting roads and obstacles in radar images. *Vehicular Technology, IEEE Transactions on*, 50(1):170–182, Enero 2001.
- [4] Massimo Bertozzi, Alberto Broggi, and Alessandra Fascioli. A stereo vision system for real-time automotive obstacle detection. *In International Conference on Image Processing*, volume 2, pages 681 – 684, Septiembre 1996.
- [5] S. Nedevschi, R. Danescu, D. Frentiu, T. Marita, F. Oniga, C. Pocol, R. Schmidt, and T. Graf. High accuracy stereo vision system for far distance obstacle detection. *In Intelligent Vehicles Symposium, 2004 IEEE*, pages 292 – 297, Junio 2004.
- [6] Ioannis Kostavelis, Lazaros Nalpantidis, and Antonios Gasteratos. Comparative presentation of real-time obstacle avoidance algorithms using solely stereo vision. *In IARP/EURON International Workshop on Robotics for risky interventions and Environmental Surveillance-Maintenance*, Sheffield, UK, January 2010.
- [7] Anwar Hasni Abu Hasan, Rostam Affendi Hamzah, and Mohd Haffiz Johar. Region of interest in disparity mapping for navigation of stereo vision autonomous guided vehicle. *In Proceedings of the 2009 International Conference on Computer Technology and Development - Volume 01, ICCTD '09*, pages 98–102, Washington, DC, USA, 2009. IEEE Computer Society.
- [8] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.
- [9] Jean-Yves Bouguet. Matlab camera calibration toolbox. 2000.
- [10] Gary Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.
- [11] Gary Bradski and Adrian Kaehler. *Learning OpenCV, Computer Vision with OpenCV Library*. O'Reilly Media, first edition, 2008.
- [12] Andreas Geiger, Martin Roser, and Raquel Urtasun. Efficient large-scale stereo matching. *In Asian Conference on Computer Vision*, Queenstown, New Zealand, November 2010.
- [13] Pablo de Cristóforis, Sol Pedre, Andrés Stoliar, and Javier Caccavelli. Exabot: a mini robot for research, education and popularization of science. *VI Latin American Summer School in Computational Intelligence and Robotics - EVIC2009*, december 2009.
- [14] Evasión de obstáculos (video): <http://youtu.be/Gxw10E4BhbQ>.