

# Navegación autónoma para robots móviles usando visión estéreo

Cintia Corti, Carlos Augusto Lyon Di Pietro, Facundo Pessacg, Kevin Allekotte

**Resumen**—Como trabajo final de la materia VISIÓN EN ROBÓTICA de DC.UBA.AR presentamos un método para la navegación autónoma de robots usando visión estéreo. La implementación se realizó para el EXABOT usando las librerías OPENCV y LIBELAS.

## I. INTRODUCCIÓN

En este trabajo presentamos un método para la navegación autónoma de un robot basado en algoritmos de computer vision usando cámaras en estereo.

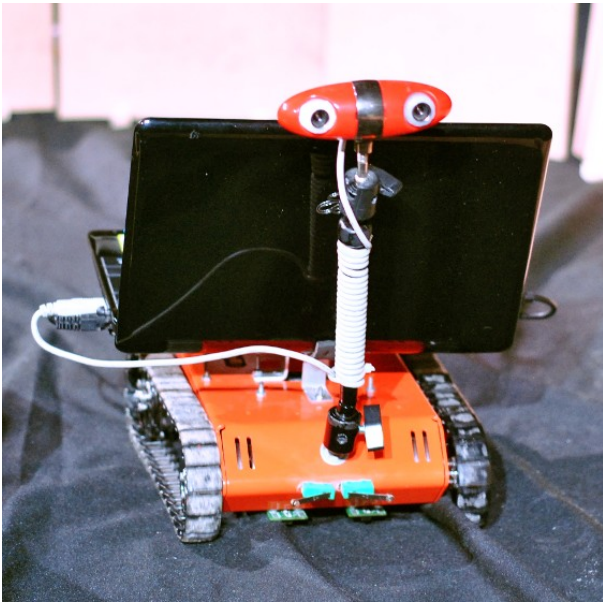


Figura 1. Exabot montado con cámara estereo y netbook.

La visión en estéreo consiste de 2 cámaras alineadas en la misma dirección que toman las mismas imágenes desde puntos apartados ligeramente, asemejándose al sistema de visión de los humanos y animales. La idea es alinear las imágenes (previamente rectificadas) y calcular la disparidad en cada punto. Cada punto del ambiente es captado por las 2 cámaras, y por propiedades trigonométricas vemos que la distancia del punto a las cámaras es proporcional a la diferencia de posición horizontal observado entre las cámaras. Así, un punto lejano va a proyectarse en las 2 imágenes en casi la misma posición, y un punto cercano a las cámaras va a proyectarse en posiciones distintas en cada cámara. Gracias a esta propiedad y calculando el mapa de disparidad entre las 2 imágenes podemos obtener una estimación muy buena de la geometría del espacio frente a las cámaras. Un detalle no menor es

que para calcular el mapa de disparidad de forma precisa las imágenes de las cámaras tienen que estar cuidadosamente rectificadas y alineadas. La rectificación corrige errores de deformación de las lentes para que las imágenes representen mas fielmente la proyección del espacio al plano de la cámara, y la alineación corrige la dirección en la que apuntan las cámaras para tener un sistema de referencia certero.

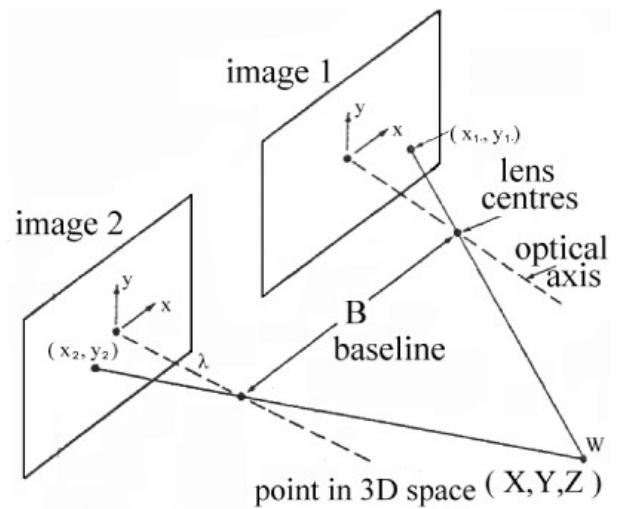


Figure 2. Stereo image geometric model.

El objetivo del método es permitir a un robot la navegación autónoma, por lo que es importante la performance para poder correr en tiempo real y proveer acciones correspondientes basado en la visión. Vamos a usar las librerías de OpenCV para procesar los streams de video y LibElas para calcular el mapa de disparidad.

Para llevar a cabo una navegación autónoma se debe implementar una estrategia basada en la observación del mundo (cámaras, mapa de disparidad) que decida acciones a ser llevadas a cabo por los actuadores (ruedas del exabot). Buscamos una estrategia que evite la colisión con objetos y avance hacia espacios libres del ambiente.

## II. MÉTODO PROPUESTO

### II-A. Calibración

Las lentes de las cámaras tienen distorsiones que provocan que las imágenes captadas no representen exactamente la proyección de los objetos en el plano de captura. Por esto, y para obtener luego un mapa de disparidad preciso, es necesario aplicar las correcciones necesarias para “rectificar” las imágenes.

Para esto utilizamos software en el entorno Matlab, que permite calcular los parámetros intrínsecos y extrínsecos de las cámaras. Sacamos fotos estereoscópicas de una grilla -un damero- de dimensiones conocidas en distintas posiciones. Se indican características como origen y tamaño de la grilla, y el programa calcula las propiedades de la cámara como distancia focal, puntos principales, skew, distorsión, rotación y distancia entre las cámaras. Con estos parámetros podemos rectificar futuras capturas de la cámara para obtener imágenes alineadas y listas para procesar.



Figura 2. Imágenes de prueba para calcular los parámetros de rectificación

### II-B. Mapa de Disparidad

Con las imágenes correctamente rectificadas podemos usarlas para calcular el mapa de disparidad. Para esto vamos a usar la librería LibElaS –Library for Efficient Large-scale Stereo Matching– desarrollada por Andreas Geiger (Karlsruhe Institute of Technology).

En la figura 3 podemos ver primero una imagen tal cual es captada por una de las cámaras, luego las 2 imágenes superpuestas (rectificadas y alineadas), y la tercera es el mapa de disparidad que obtenemos de la librería elas.

Este mapa de disparidad fue coloreado para entenderlo mejor visualmente, pero es simplemente un campo escalar donde la “intensidad” de un pixel representa la cercanía de un objeto en ese punto. Observamos que hay zonas en negro, donde no tenemos información de la profundidad. Esto puede deberse a que no hay suficiente contraste en esas partes de la imagen, o que el algoritmo no encontró matches entre las 2.



Figura 3. Una cámara; rectificadas y alineadas; mapa de disparidad

Hay que tener esto en cuenta a la hora de usar el mapa, pero igualmente es usable para nuestro propósito de navegación autónoma.

### II-C. Parametrización de atributos

Para que el mapa de disparidad recién calculado sea útil, es necesario extraer información de éste que pueda usarse en la decisión de las acciones a tomar.

Dado que las acciones razonables que puede realizar un robot como el exabot para navegar ambientes son: avanzar,

avanzar hacia la izquierda, avanzar hacia la derecha o girar en el lugar, vamos a analizar la presencia de obstáculos en alguna de estas direcciones.

Definimos 3 áreas disjuntas del campo de vision del robot –izquierda, centro y derecha– y nuestro método se basa en decidir si hay espacio para avanzar en estas direcciones. Descartamos un porcentaje (10 %) del borde de la imagen que consideramos que tiene ruido, especialmente luego de rectificar y alinear las imágenes, y la dividimos horizontalmente en 3 áreas iguales. A cada una de estas áreas le calculamos el promedio de la intensidad de los pixels, llamamos  $p_{izq}$ ,  $p_{med}$  y  $p_{der}$ .

Esto nos da una idea del espacio que hay frente al robot en estas regiones. Si el promedio es relativamente alto, significa que hay puntos del ambiente que estan cerca de la cámara, y por lo tanto, que hay obstáculos en esa dirección. Si por el contrario el promedio es un valor bajo, podemos asumir que hay espacio libre para avanzar. Definimos arbitrariamente 2 umbrales para los valores de los promedios. Si un valor supera el umbral alto consideramos que hay un obstáculo muy cercano y que no es seguro avanzar en esa dirección; si está entre los 2 umbrales hay algo frente al robot pero no es crítico; y si esta por debajo del umbral bajo consideramos que el camino está libre.

Además, para tener estimaciones más precisas de los valores reales, promediamos cada valor también en el tiempo. Es decir, calculamos  $p_{izq}$ ,  $p_{med}$  y  $p_{der}$  para cada frame durante un tiempo determinado, y luego promediamos el valor de cada área.

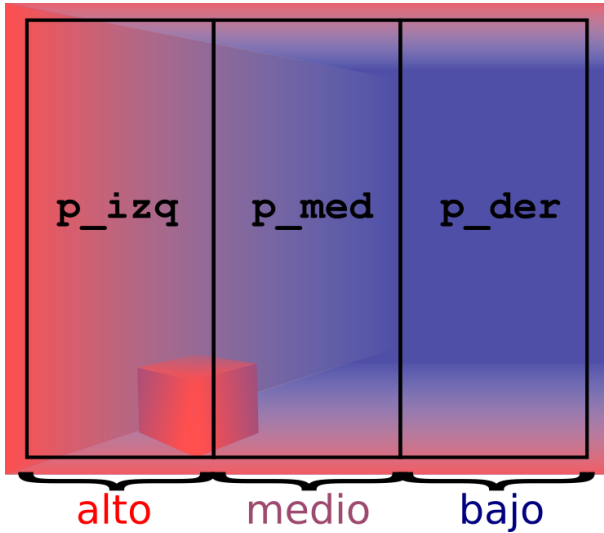


Figura 4. Análisis del mapa de disparidad

En resumen, cada cierta cantidad de frames obtenemos estimaciones de la cercanía de obstáculos para cada una de las 3 áreas. Esto nos identifica 27 posibles escenarios para tomar decisiones en cada instante. Elegimos la cantidad de frames a promediar en función de los FPS para tomar una decisión cada aproximadamente 3 segundos.

## II-D. Estrategia

Para llevar a cabo la navegación necesitamos una estrategia que determine qué acciones tomar ante cada escenario posible. Usamos la categorización de posibilidades que mencionamos recién, y vamos a enumerar las acciones a tomar en cada caso.

En la siguiente tabla [5] se presenta la acción a tomar según los valores de los promedios de las áreas. La estrategia está pensada para que explore lo máximo posible sin colisionar con obstáculos.

$p_{izq}$	$p_{med}$	$p_{der}$	Acción
bajo	bajo	bajo	avanzar rápido
bajo	bajo	medio	
medio	bajo	bajo	
medio	bajo	medio	
bajo	bajo	alto	avanzar hacia la izquierda
medio	bajo	alto	
alto	bajo	bajo	avanzar hacia la derecha
alto	bajo	medio	
alto	bajo	alto	avanzar lento
bajo	medio	bajo	avanzar lento
bajo	medio	medio	avanzar lento hacia la izquierda
bajo	medio	alto	
medio	medio	bajo	avanzar lento hacia la derecha
alto	medio	bajo	
alto	medio	alto	avanzar muy lento
medio	medio	medio	
alto	medio	medio	
medio	medio	alto	
bajo	alto	bajo	girar
medio	alto	medio	
alto	alto	alto	
bajo	alto	medio	girar a la izquierda
bajo	alto	alto	
medio	alto	alto	
medio	alto	bajo	girar a la derecha
alto	alto	bajo	
alto	alto	medio	

Figura 5. Estrategia de navegación

Observar que la acción a tomar en cada caso es determinística y sólo depende del estado actual del sensado (o de los últimos  $\sim 3$  segundos de sensado).

Por ejemplo, en el caso de la figura 4 (donde los valores de los promedios son *alto*, *medio*, *bajo*), la estrategia dice que hay que avanzar lento hacia la derecha.

Cada una de estas acciones se traduce a un par de velocidades para los motores que producen que el robot haga esa acción. En el ejemplo, “avanzar lento hacia la derecha” se traduce a (0.216, 0.072), osea una velocidad de 0.216 para el motor izquierdo y 0.072 para el derecho. El efecto producido es que el exabot avanza levemente mientras rota hacia la derecha.



### III. EXPERIMENTOS

Para llegar a este método de navegación autónoma experimentamos con diferentes opciones y elegimos la mejor. Además hay una serie de parámetros a optimizar para mejorar el resultado.

#### III-A. Toma de imágenes

Los parámetros variables en esta instancia son el tamaño y tipo de la imagen y el brillo. Como no nos interesa mucho la granularidad del mapa de disparidad (porque de todas formas vamos a promediar grandes áreas) decidimos trabajar con un tamaño de imagen chico (200 o 300 pixeles de ancho) para que el procesamiento sea más rápido y las convertimos en escala de grises. El brillo lo ajustamos luego de hacer un par de pruebas para que funcione bien en interiores iluminados, y se puede cambiar si cambian las condiciones.

#### III-B. Alineación, rectificación

Los parámetros de rectificación los obtenemos de un archivo de configuración previamente creado con los datos obtenidos de la calibración para esa cámara. La alineación horizontal se ajusta en tiempo de ejecución porque puede variar entre los experimentos. En este paso también recortamos un 8~10% de los bordes para descartar las imperfecciones resultantes de rectificar, girar y alinear las imágenes.

#### III-C. Mapa de Disparidad y parametrización del escenario

El mapa de disparidad contiene información de la disparidad entre las imágenes, que se corresponde con algún factor a la cercanía de obstáculos en el ambiente.

Mencionamos en el Método que definimos 2 umbrales para el valor de las disparidades. Ajustamos estas constantes de forma aproximada y luego de algunas pruebas, como para que valores *bajos* se correspondan con más de un metro de distancia a un obstáculo, valores *medios* con obstáculos que se encuentran entre los 40cm y 1m, y valores *altos* con obstáculos a menos de 40cm. De esta forma durante la navegación se acerca suficientemente a las paredes como para explorar, pero evita las colisiones con una distancia segura.

También experimentamos con distintos valores del promedio temporal y finalmente decidimos 8 frames. A la velocidad que procesamos los datos, este valor es suficiente como para tomar una decisión cada aproximadamente 3 o 4 segundos.

#### III-D. Estrategia

Una vez categorizados los 27 escenarios posibles, decidimos la estrategia para cada caso o conjunto de casos basándonos en el sentido común. En general el método es el de avanzar hacia espacios vacíos y alejándose de obstáculos. Definimos velocidades distintas para cuando hay espacio y para valores medios de forma de explorar rápido pero con cuidado cuando hay peligro de colisión.

Hay muchas estrategias diferentes que se pueden implementar, y cada una provocará una navegación diferente. La que elegimos creemos que es adecuada para navegar interiores con paredes con el exabot.

### IV. CONCLUSIONES

El uso de cámaras estereoscópicas para reconstruir la geometría de un ambiente se asemeja mucho a la forma en que lo hacen humanos y otros animales. La disparidad entre las imágenes brinda información muy valiosa acerca de la distancia a objetos y es muy útil para la navegación.

En principio pareciera que el uso de 2 cámaras no diferiría mucho de una sola cámara, pero con estos resultados se puede observar la utilidad del método. Un mecanismo que provee un mapa de distancias del espacio frente al robot es sin duda uno de los mecanismos más valiosos de sentido que se pueden implementar, especialmente para navegación autónoma.

Pudimos implementar este método en un robot relativamente simple y con hardware accesible, y una estrategia de navegación autónoma que permite una buena exploración. Se puede mejorar la precisión y performance de la obtención del mapa de disparidad y mejorar la estrategia y hacer un robot más reactivo al ambiente.

Existen también muchos otros métodos de extracción de información a partir de la disparidad, pero la separación en 3 zonas es una forma simple que funciona sorprendentemente bien para este caso de uso.

Para trabajo futuro se puede combinar este método con mecanismos de localization and mapping para hacer una navegación más inteligente y exhaustiva, o también aumentar la cantidad de dimensiones o motion planning o combinar con otras formas de sentido, entre otras.



### V. BIBLIOGRAFÍA

**LIBELAS:** Librería para computar mapas de disparidad a partir de pares de imágenes en escala de grises rectificadas. <http://www.cvlibs.net/software/libelas.html>

**OPENCV:** Librería de funciones para computer vision en tiempo real. <http://opencv.willowgarage.com/>