

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.

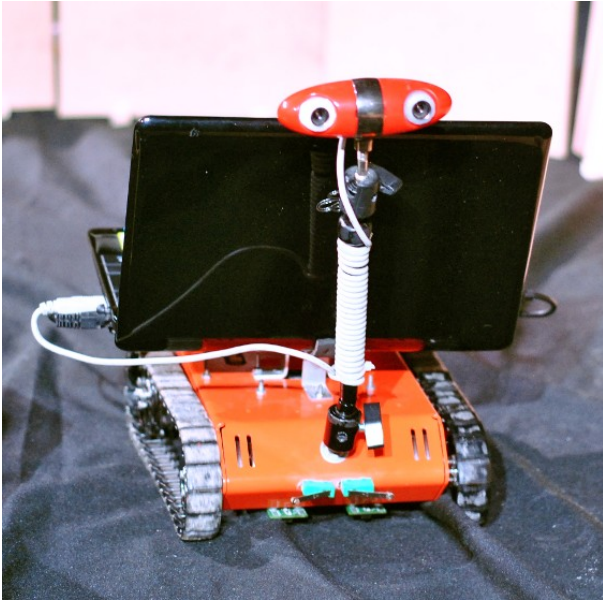


Figura 1. Exabot montado con cámara estereo y netbook.

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.

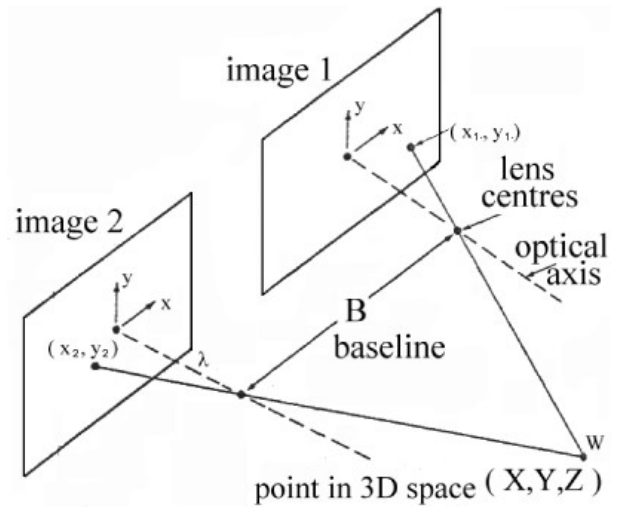


Figure 2. Stereo image geometric model.

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.

III. EXPERIMENTOS

IV. CONCLUSIONES

V. BIBLIOGRAFÍA

LIBELAS: Librería para computar mapas de disparidad a partir de pares de imagenes 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/>

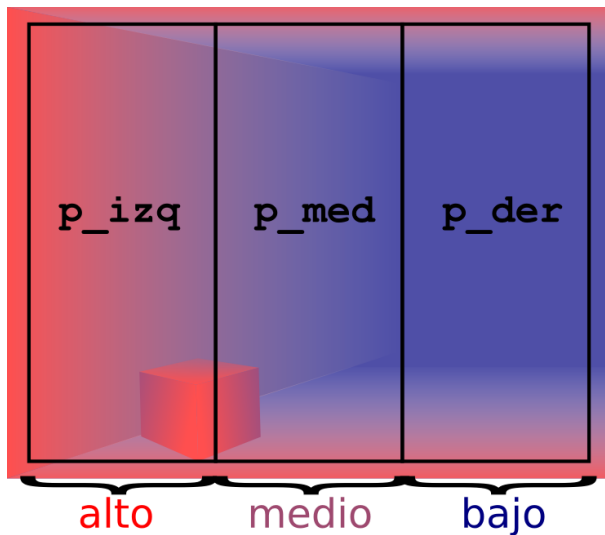


Figura 4. Análisis del mapa de disparidad

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.