



Mapeo denso en tiempo real sobre sistemas SLAM basados en visión estéreo

6 de febrero de 2017

Ariel Roberto D'Alessandro
ariel.dalessandro@gmail.com

Director
Pire, Taihú
tpire@dc.uba.ar

Co-Director
Baravalle, Rodrigo
baravalle@cifasis-conicet.gov.ar



Facultad de Ciencias Exactas, Ingeniería y
Agrimensura
Universidad de Rosario
Av. Pellegrini 250 - Planta Baja - (S2000BTP)
Rosario - Rep. Argentina.
Teléfono: +54 - 341 - 4802649/52 - interno 112
<http://www.fceia.unr.edu.ar>

Reconstrucción densa en tiempo real sobre sistemas SLAM basados en visión estéreo

Uno de los principales problemas presentes en aplicaciones relativas a robots autónomos móviles consiste en el planeamiento de una trayectoria segura hacia un objetivo (u objetivos) que el robot debe alcanzar. Para esto, resulta necesario contar con una representación del ambiente que se está transitando (mapa) a medida que el robot se desplaza.

Dentro de este requerimiento pueden distinguirse dos cuestiones a resolver:

1. El mapa debe ser extendido y actualizado en tiempo real para que el robot pueda refinar su trayectoria a medida que explora el ambiente.
2. El mapa debe ser suficientemente detallado (denso) para que el robot pueda realizar un planeamiento seguro, evitando posibles obstáculos.

El primer problema es conocido como *SLAM* (Simultaneous Localization And Mapping). Dentro del estado del arte actual pueden encontrarse diversas soluciones propuestas, e.g. S-PTAM, que estiman una localización fiable del robot aunque el mapa generado es esparso, resultando no apto para planificación de trayectorias y detección de obstáculos.

El presente trabajo aborda el problema de reconstrucción densa en sistemas de SLAM que utilizan cámaras estéreo como sensor principal. La solución propuesta puede dividirse en dos etapas capaces de ser ejecutadas en paralelo: generación y reproyección 3D de mapas de disparidad; transformación y fusión global de nube de puntos.

Para generar los mapas de disparidad se utiliza la librería de código abierto LIBELAS, disponible públicamente, que provee un método eficiente con interpolación de áreas de bajo gradiente. Estos mapas de disparidad son reproyectados para obtener una nube de puntos 3D local.

Utilizando la pose estimada por el sistema SLAM, la nube de puntos es transformada y continuamente refinada sobre el sistema de coordenadas global (mapa). Previamente, haciendo uso de un mecanismo similar al propuesto en StereoScan, se fusionan puntos redundantes y descartan outliers para reducir la complejidad de la nube de puntos y afinar la representación del entorno.

La solución propuesta se implementó como un nodo del framework ROS (*Robot Operating System*) para ser ejecutado en paralelo y conjuntamente con el sistema S-PTAM (*Stereo Parallel Tracking And Mapping*) de código abierto. Los experimentos realizados con *datasets* de dominio público muestran que el sistema es apto para generar en tiempo real un mapeo denso, en entornos de grandes dimensiones, sin requerir el uso de GPUs.

Índice general

1. Introducción	5
1.1. Motivación y definición del problema	5
1.2. Objetivos	5
1.3. Organización del trabajo	5
2. Trabajo relacionado	7
2.1. Reconstrucción 3D	7
2.2. SLAM - Localización y mapeo simultáneo	7
2.2.1. V-SLAM - Localización y mapeo simultáneo visual	8
2.2.2. S-PTAM	8
2.3. V-SLAM denso	9
2.3.1. Densificación de métodos esparsos	9
2.3.1.1. OpenCV	9
2.3.1.2. LIBELAS	10
2.3.2. Métodos directos	11
2.3.3. Métodos basados en superpíxeles	12
2.3.4. Métodos semánticos	13
3. Cámaras como sensores	15
3.1. Modelo y geometría de una cámara monocular	15
3.2. Modelo y geometría de cámaras estéreo	16
3.2.1. Correspondencia entre puntos	17
3.2.2. Rectificación estéreo	17
3.2.3. Triangulación de puntos del espacio	18
4. Método propuesto de mapeo denso basado en disparidad	21
4.1. Esquema general	21
4.2. Cálculo de mapas de disparidad	22
4.2.1. OpenCV	22
4.2.2. LIBELAS	22
4.3. Proyección al plano de la imagen	24
4.3.1. Frustum of view	24
4.3.2. Proyección al plano de la imagen - Obtención del depth map	25
4.4. Consistencia y redundancia (StereoScan)	26
4.4.1. Heurística de detección de outliers y fusión	26
4.5. Detalles de implementación	26

5. Experimentación y resultados	27
6. Conclusiones	29
6.1. Trabajo futuro	29

Capítulo 1

Introducción

1.1. Motivación y definición del problema

1.2. Objetivos

1.3. Organización del trabajo

Capítulo 2

Trabajo relacionado

La presente tesina investiga y propone un método de reconstrucción 3D o mapeo denso en tiempo real sobre sistemas de localización y mapeo simultáneo (SLAM) basados en visión estéreo. En esta sección, se exponen brevemente los trabajos relacionados, describiendo la naturaleza del problema que abordan.

↑
al problema de mapeo denso en SLAM???

2.1. Reconstrucción 3D

La reconstrucción 3D se define como proceso mediante el cual se captura y representa la forma y apariencia de entornos u objetos presentes en la realidad.

Ha sido un tópico central en visión computarizada durante décadas aunque también es una línea de investigación activa en diversos campos como el diseño asistido por computadoras, imágenes médicas, realidad virtual y extendida, robótica móvil, entre otros. Numerosos métodos han sido propuestos para abordar las variantes de este problema de diferentes naturalezas según el área de interés y los sensores utilizados para obtener información del ambiente.

Dado el enfoque de la presente tesina, resultan de interés principalmente aquellos métodos de reconstrucción 3D que utilizan cámaras como sensores. En la última década numerosos trabajos se han publicado en este área, como pueden verse clasificados y comparados en [1, 2, 3].

esta

2.2. SLAM - Localización y mapeo simultáneo

Complementariamente al problema de reconstrucción 3D, particularmente en el campo de la robótica móvil, se plantea el problema de localización y mapeo simultáneo, conocido como SLAM (Simultaneous Localisation and Mapping). El desafío consiste en construir incrementalmente un mapa consistente de un entorno desconocido mientras se determina la posición y orientación del agente (robot).

el problema de SLAM

poner citas de trabajos con encoders y cámaras

en inglés
y aclarar en español. En las últimas décadas, ha sido extensivamente estudiado utilizando diferentes sensores: IMUs (unidades de medición inercial), sensores de tipo SONAR [4], sensores infrarrojos [5], escáneres láser [6], GPS (sistema de posicionamiento global), codificadores rotatorios y cámaras. El interés sobre el problema de SLAM ha crecido en gran medida dado que al solucionarlo se proveen herramientas necesarias para que los robots móviles puedan navegar de manera autónoma.

Una introducción a Localización y mapeo simultáneo (SLAM), descripción del problema y soluciones esenciales pueden encontrarse en [7, 8].

al mismo

2.2.1. V-SLAM - Localización y mapeo simultáneo visual

Recientemente ha habido un creciente interés en sistemas de SLAM visual, que utilizan cámaras como sensor principal, debido a las importantes ventajas que ~~estos~~ presentan: bajo costo y menor consumo energético en comparación con otros sensores como escáneres laser y su alta disponibilidad en dispositivos móviles. Las cámaras son sensores pasivos por lo que no interfieren ~~entre sí~~ y pueden utilizarse en entornos cerrados donde el uso de GPS puede verse imposibilitado. Asimismo, son menos restrictivos que los ~~codificadores rotatorios~~, limitados a robots terrestres y altamente imprecisos en terrenos irregulares. ~~encoders~~ ~~los cuales se encuentran~~

~~de SLAM visual~~
~~La mayoría de los métodos utilizan secuencias de imágenes 2D o video 2D como entrada del sistema~~, que es procesada para estimar la profundidad de la escena capturada, reconstruyendo su estructura 3D [9, 10, 11, 12]. A partir de la detección y seguimiento de marcas naturales del ambiente (*landmarks*), ~~la mayoría de los sistemas de SLAM visual estiman tanto la posición del robot como la ubicación de las marcas en el entorno. El mapa se construye con las estimaciones de las posiciones de dichas marcas, las cuales van siendo ajustadas a medida que son observadas desde distintas posiciones.~~

~~los métodos de SLAM visual basados en la extracción de características visuales en las imágenes (features)~~
~~Las técnicas existentes~~ son capaces de estimar con precisión el desplazamiento de la cámara ~~del entorno~~ ~~el mapa reconstruido~~ y computar un mapa esparsa en tiempo real [13, 14, 15, 16, 17, 18, 19]. Sin embargo, ~~este~~ contiene solamente una pequeña porción de los puntos 3D de la escena y ~~una reconstrucción completa requiere conocer la profundidad de cada uno de ellos~~

ESTA SECCIÓN VA EN EL CAPÍTULO DE MÉTODO PROPUESTO

2.2.2. S-PTAM

llamado S-PTAM (Stereo Parallel Tracking and Mapping)

En [20] se presenta un sistema SLAM basado en visión estéreo, capaz de estimar la localización de un robot móvil en tiempo real en trayectorias de gran longitud. S-PTAM se inspira en PTAM [21] aprovechando la capacidad de cómputo de unidades de procesamiento paralelo, dividiendo el problema al igual que PTAM en dos tareas principales: seguimiento de la cámara y la construcción del mapa.

S-PTAM divide las tareas de localización y mapeo en dos hilos de ejecución diferentes, aprovechando la capacidad de cómputo de procesadores con dos o más núcleos.

Si bien el método de reconstrucción densa propuesto en esta tesina no se restringe a ser ejecutado sobre un sistema de SLAM en particular, fue diseñado tomando en consideración la interfaz y el funcionamiento de S-PTAM. La Figura 2.2.1 muestra un escenario de ejemplo, donde se utiliza S-PTAM para la estimación de la trayectoria.

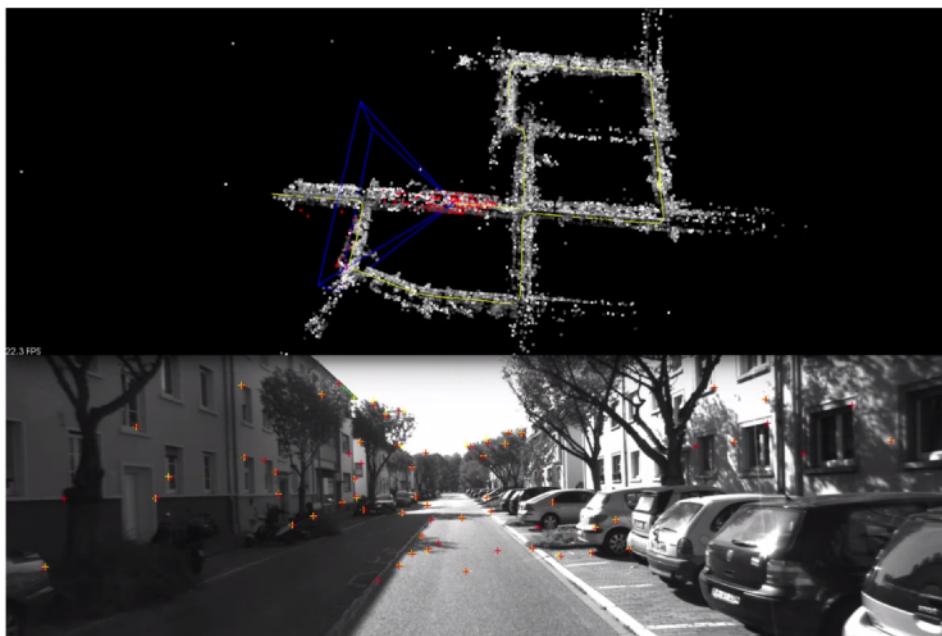


Fig. 2.2.1: Reconstrucción del ambiente (mapa) y trayectoria estimada por S-PTAM durante el procesamiento de una secuencia de imágenes obtenida del recorrido de un vehículo.

2.3. V-SLAM denso

Para obtener un mapa denso a partir de información visual se debe resolver el problema de correspondencia, es decir asociar cada pixel de una imagen con su correspondiente en otra. Sin embargo, en las regiones con poca textura -bajo gradiente en la intensidad- debido a que... resulta complejo es difícil realizar esta correspondencia píxel-a-píxel, problema denominado como “el muro blanco”. A su vez, pueden existir occlusiones entre las imágenes o variaciones debido a la especularidad de los objetos a medida que la cámara se desplaza, dificultando aún más la asociación píxel-a-píxel.

El enfoque tradicional, conocido como búsqueda sobre la línea epipolar, impone ciertas restricciones geométricas y reduce la complejidad de la búsqueda en 2D a 1D, aunque no soluciona los problemas de asociación antes mencionados. A continuación se clasifican algunos enfoques que han abordado la generación de mapas densos a partir de información visual.

2.3.1. Densificación de métodos esparsos

El problema de asociación densa de píxeles entre imágenes ha sido abordado en la última década por diversos trabajos^[22, 23] mostrando que es posible aumentar la densidad de los mapas a partir de información visual. Particularmente, el interés en los mapas de disparidad ha crecido con la introducción de cámaras estéreo.

Esta sección de las librerías la pondría en el capítulo de Cámaras como sensores o Trabajo Propuesto o Resultados...

2.3.1.1. OpenCV

OpenCV es una biblioteca libre de visión artificial originalmente desarrollada por Intel, que provee un método de correspondencia estéreo llamado “Block matching (BM) algorithm”.

Este algoritmo, similar al algoritmo desarrollado por Kurt Konolige^[24], funciona utilizando pequeñas ventanas de “sumas de diferencia absoluta” (SAD) para encontrar puntos

que suman la diferencia absoluta entre los píxeles comprendidos dentro de la ventana???

siempre tilizar texto en itálica cuando se usan palabras en inglés

explicar... quizás sea más fácil poner una cuenta

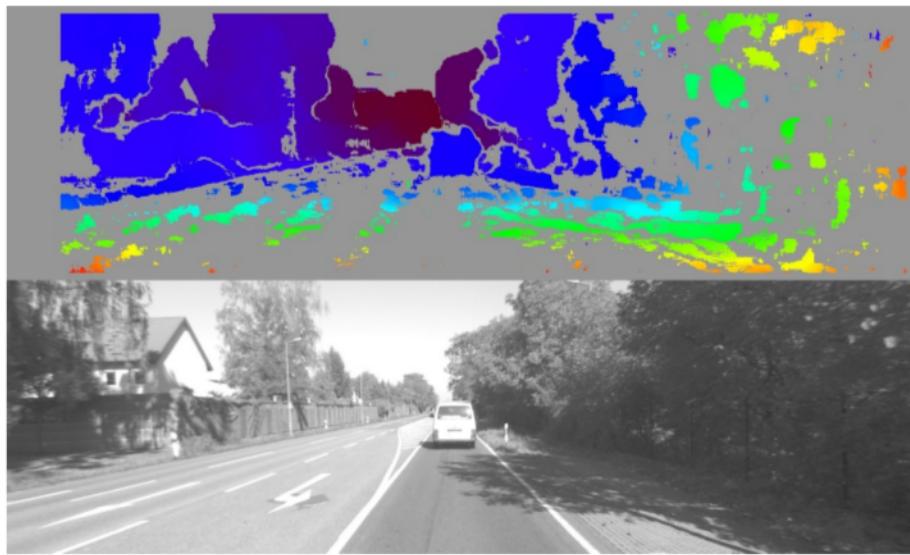


Fig. 2.3.1: Arriba: Mapa de disparidad calculado mediante OpenCV StereoBM. Abajo: Imagen izquierda perteneciente al dataset KITTI secuencia 04.

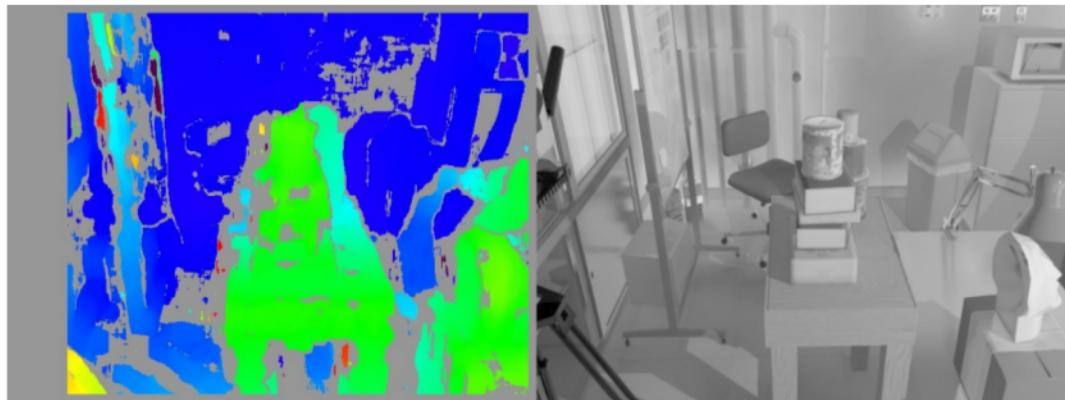


Fig. 2.3.2 **Arriba**: Mapa de disparidad calculado mediante OpenCV StereoBM. **Abajo**: Imagen izquierda perteneciente al dataset Tsukuba Daylight.

coincidentes entre el par de imágenes estéreo rectificadas.

Si bien este método es muy rápido y efectivo, solo encuentra aquellos puntos que presentan alta textura en ambas imágenes, por lo que puede resultar inadecuado para entornos complejos como puede distinguirse en [2.3.1](#) y [2.3.2](#).

2.3.1.2. LIBELAS

LIBELAS (Library for Efficient Large-scale Stereo Matching) [\[25\]](#) es una librería multiplataforma de código abierto para computar mapas de disparidad a partir de pares de imágenes estéreo rectificadas de alta resolución.

El método empleado se basa en que, pese al hecho de que muchas correspondencias estéreo son altamente ambiguas, algunas de ellas pueden ser robustamente matcheadas. Asumiendo variaciones suaves en la disparidad es posible sectorizar la imagen, mediante la triangulación

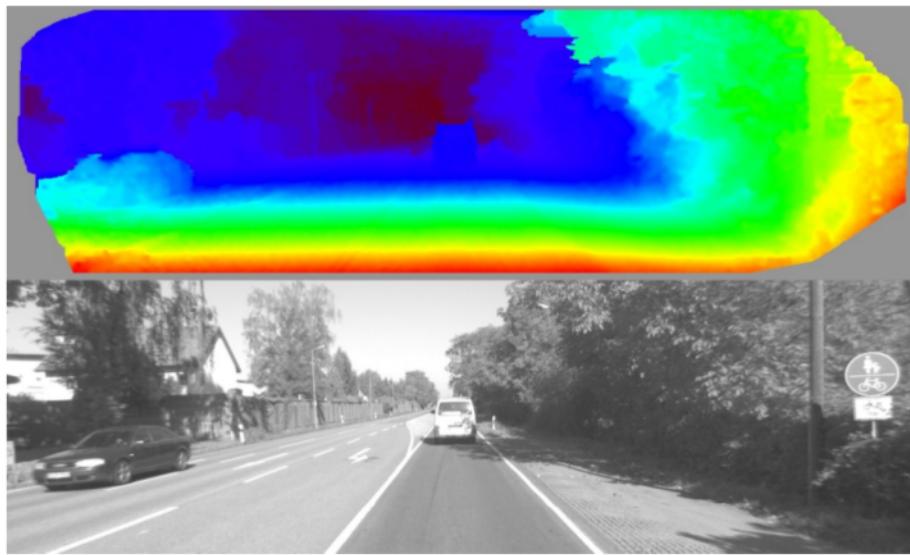


Fig. 2.3.3: Arriba: Mapa de disparidad calculado mediante LibELAS. Abajo: Imagen izquierda perteneciente al dataset KITTI secuencia 04.

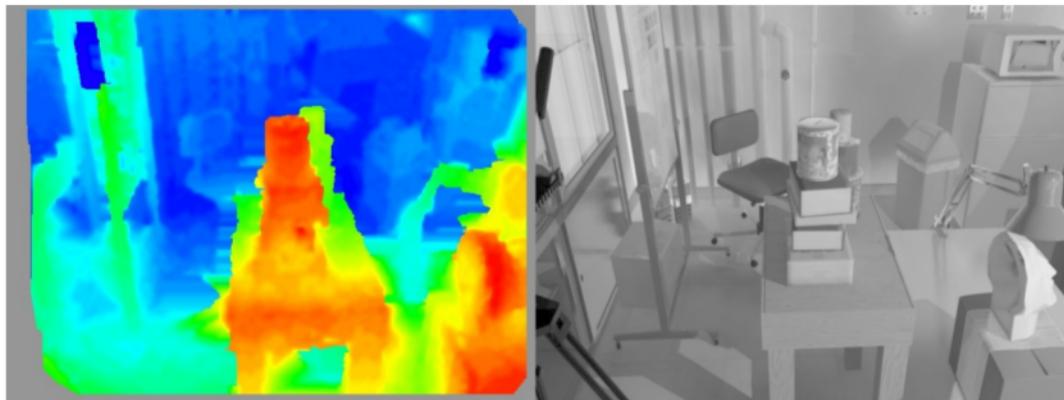


Fig. 2.3.4: Izquierda: Mapa de disparidad calculado mediante LibELAS. Derecha: Imagen izquierda perteneciente al dataset Tsukuba Daylight.

poner CITA *no usar comillas, usar itálica para nombres importante si es deseado.*
de Delaunay, sobre estos puntos de soporte para la estimación de las restantes disparidades.
En 2.3.3 y 2.3.4 pueden observarse resultados de su ejecución.

las figuras En el trabajo StereoScan [25] se parte de un sistema SLAM basado en características visuales (features) obteniendo un mapa esparsa. Los pares de imágenes provenientes de una cámara estereo son procesados para generar mapas de disparidad densos mediante LIBELAS [26], aumentando la cantidad de puntos presentes en el mapa.

No hay más citas para hacer sobre trabajos que sean esparsos y luego densifiquen?

Esto queda en trabajo relacionado

2.3.2. Métodos directos

Recientemente, una nueva línea de investigación se ha desarrollado entorno a los métodos directos [27] [28] [29], que a diferencia de los basados en features, ajustan la geometría y fotometría de la escena directamente sobre las intensidades de la imagen, utilizando todos los píxeles de alto contraste, incluyendo vértices, aristas y áreas de alta textura. Este enfoque

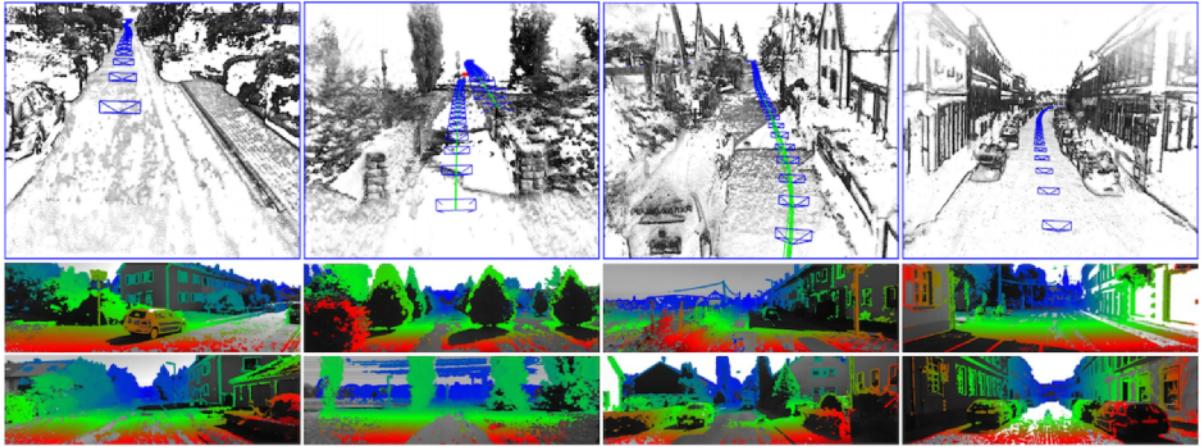


Fig. 2.3.5: Nubes de puntos y mapas de profundidad semi-densos para el dataset Kitti generados por Stereo LSD-SLAM

utiliza mayor parte de la información disponible en las imágenes, por lo que puede obtener mapas 3D con mayor densidad (densos y semi-densos).

Sin embargo, el nivel de desarrollo de SLAM directo aún es bajo en comparación con los métodos basados en características visuales, resultan ~~ménos~~^{do ser} flexibles para la eliminación de outliers y ~~la exactitud es menor, como puede verse en [30].~~ ^{preciosos} ~~preciosos~~ ^{itálica}

Uno de los trabajos más emblemáticos es LSD-SLAM [31] (Large Scale Direct SLAM) que implementa una solución para secuencias de larga escala en tiempo real en CPU (sin GPU) utilizando una cámara monocular. Posteriormente, este sistema se extiende para cámaras estéreo [32] y omnidireccionales [33].

Recientemente, en [34] se ha publicado un sistema de V-SLAM directo que fusiona información fotométrica con ~~inercial, proveniente de unidades de medición inercial (IMU).~~ ^{explayarse un poco más} mediciones de una

2.3.3. Métodos basados en superpíxeles

El enfoque predominante en SLAM se basó tradicionalmente en encontrar correspondencias en áreas de la imagen altamente texturadas -características visuales de bajo nivel-, por lo que largas regiones sin textura, usualmente presentes en entornos interiores y urbanos, son difíciles de reconstruir por estos sistemas. Para superar esta dificultad y aumentar la densidad de estas reconstrucciones, se han desarrollado métodos que explotan características visuales de mediano nivel, como los superpíxeles.

Los superpíxeles consisten en regiones de la imagen con textura homogénea. A diferencia de las características visuales de bajo nivel, presentan ~~algunos~~ inconvenientes como son la baja repetibilidad y la alta dependencia a las especularidades de los objetos. ^{? detallar esto}

En [35] se segmentan las imágenes en superpíxeles y se asume que cada región corresponde a una superficie planar, cuyos contornos son ~~proyectados~~ y mapeados a lo largo de la secuencia. El sistema se integra a PTAM [21] para aumentar el mapa producido en tiempo real. De manera similar, DPPTAM [36] utiliza información semi-densa de la imagen para mapear el contorno de los superpíxeles como parches (superficies planares) en el entorno 3D. ^{explicar un poco mejor}



Fig. 2.3.6: Segmentación de una imagen en superpíxeles con dos niveles de granularidad diferentes.

2.3.4. Métodos semánticos

Recientemente, en SLAM visual se ha comenzado a utilizar características visuales de alto nivel, que permite reconocer y mapear objetos. Este área de investigación se encuentra altamente relacionada a tópicos de aprendizaje automatizado e inteligencia artificial. *dicho por qué se encuentra relacionada?*

Dentro del campo de SLAM visual monocular, en [37] se logra reconocer un conjunto de objetos conocidos en el mapa a través de los descriptores de bajo nivel (features) que presenta. En SLAM++ [38] se profundiza el enfoque usando directamente los objetos conocidos como los descriptores del mapa y en [39, 40] la estructura de una habitación es utilizada como descriptores de alto nivel.

En [41] se utiliza conocimiento previo sobre la forma de distintas categorías de objetos para clasificarlos y refinar su reconstrucción 3D. Recientemente, apuntando a reconstrucciones incrementales de larga escala, se publicó el trabajo [42] que realiza segmentación semántica en tiempo real para distintas categorías como se ilustra en 2.3.7.

la Fig.

Siempre referenciar las figuras de la siguiente manera:
Fig. numero_de_figura

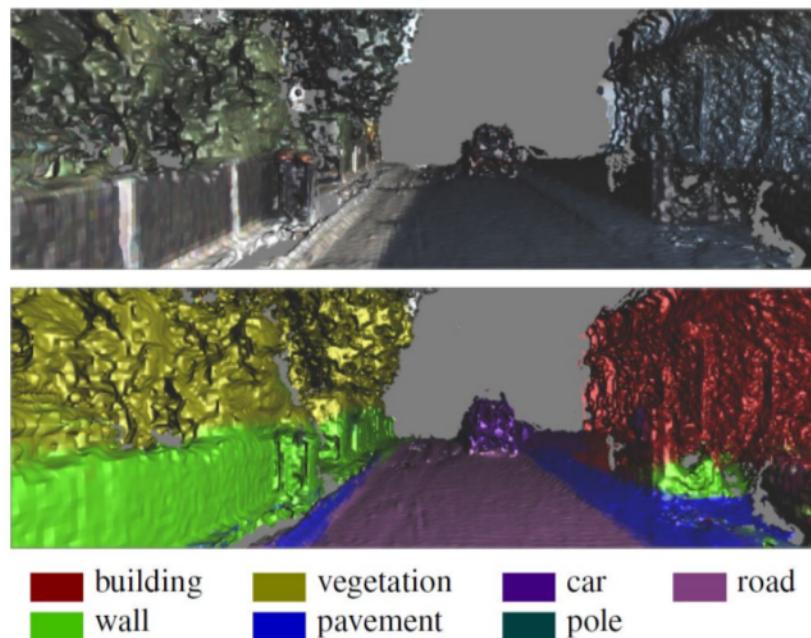


Fig. 2.3.7: Reconstrucción 3D con segmentación semántica producido por el método propuesto en [poner la CITA](#).
~~Incremental Dense Semantic Stereo Fusion for Large Scale Semantic Scene Reconstruction~~

Siempre decir de dónde obtuviste las imágenes. Por ejemplo, decir: imagen adaptada de [CITA] en caso de que hayas hecho modificaciones o imagen recuperada de [CITA] en caso de que sea la misma.

Capítulo 3

Cámaras como sensores

En el último tiempo, el desarrollo de la robótica móvil y de la visión por computadora han dado lugar a una nueva disciplina conocida como visión robótica (*robot vision*) que propone utilizar cámaras como los sensores primarios para capturar la información del ambiente. Las cámaras poseen varias ventajas por sobre otros sensores. Para empezar son mucho más económicas, de bajo consumo y fáciles de montar en robots móviles que otros sensores típicos como los láseres. Las nuevas generaciones de cámaras digitales son cada vez más asequibles y más pequeñas, y pueden proporcionar datos de alta resolución en tiempo real y rangos de medición virtualmente ilimitados. Además, las cámaras son sensores pasivos, por tanto, no interfieren unos con otros como lo hacen los sensores activos. Los requerimientos de cómputo para ejecutar los algoritmos de procesamiento de imágenes a bordo del robot móvil pueden satisfacerse en la actualidad gracias a las unidades de procesamiento disponibles. Por lo tanto, se puede afirmar que las cámaras a bordo de los robots móviles se han transformado en una parte estándar en el sistema de sensado y percepción de cualquier robot móvil, para reunir información detallada y en tiempo real sobre el entorno.

El modelo de sensado en el caso de las cámaras consiste básicamente en un mapeo entre el entorno tridimensional y el plano de la imagen bidimensional. Dependiendo el tipo de cámara (monocular, estéreo, omnidireccional, etc.) es posible utilizar diferentes modelos matemáticos que permitan relacionar puntos del mundo con su respectiva representación en la imagen.

A continuación se exponen los detalles del modelo de cámara monocular y estéreo expuestos en [9] junto con resultados geométricos de interés.

3.1. Modelo y geometría de una cámara monocular

Se modela a la cámara de manera que los puntos del espacio sean proyectados en el plano de la imagen o plano focal como se muestra en la Figura 3.1.1. Para esto se define la calibración intrínseca de la cámara como:

$$K = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \quad (3.1.1)$$

de manera de poder, utilizando coordenadas homogéneas, realizar un mapeo de la forma $(X, Y, Z, 1)^\top \xrightarrow{K} (f\frac{X}{Z} + p_x, f\frac{Y}{Z} + p_y, 1)^\top$ por multiplicación a izquierda:

$$[K | \vec{0}](X, Y, Z, 1)^\top = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} fX + Zp_x \\ fY + Zp_y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} f\frac{X}{Z} + p_x \\ f\frac{Y}{Z} + p_y \\ 1 \end{pmatrix}.$$

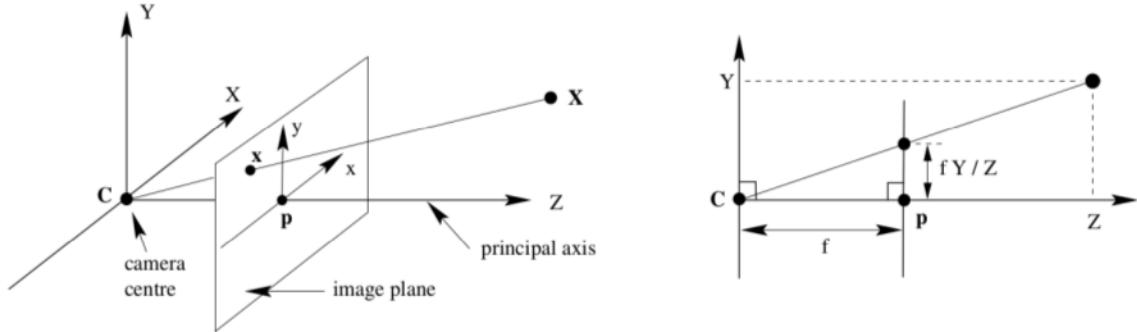


Fig. 3.1.1: Modelo geométrico *Pinhole* donde \mathbf{C} es el centro de cámara, $\mathbf{p} = (p_x, p_y)$ punto principal y la cámara se considera centrada en el origen del eje de coordenadas.

En general los puntos del ambiente son expresados en referencia al sistema de coordenadas conocido como “mundo” (*world coordinate system*), el cual esta relacionado con el sistema de coordenadas de la cámara a través de una rotación y una traslación. Sea $X_w = (X, Y, Z, 1)^\top$ un punto del espacio en referencia al sistema de coordenadas del mundo y X_c el mismo punto expresado en referencia a la cámara, es posible definir una transformación T_{cw} tal que:

$$X_c = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = T_{cw} X_w \quad (3.1.2)$$

donde R es una matriz de rotación 3×3 , t un vector de traslación 3×1 y T_{cw} pertenece al grupo de movimientos de cuerpo rígido 3D (*Lie Group, SE(3)*) [43]. De esta manera, utilizando la matriz de calibración intrínseca K , se define la matriz de proyección P que nos permitirá proyectar cualquier punto del espacio en coordenadas del mundo al plano de la imagen:

$$P = K[R|t] \quad (3.1.3)$$

La matriz P posee 9 grados de libertad (3 por K , 3 por R y 3 por t) y

$$x = P X_w \quad (3.1.4)$$

donde x es la proyección en coordenadas homogéneas (3×1) del punto X_w .

3.2. Modelo y geometría de cámaras estéreo

La geometría proyectiva entre dos cámaras es conocida como geometría epipolar (*epipolar geometry*). Esta es independiente de la escena observada, y depende únicamente de los pa-

rámetros internos y poses relativas de las cámaras involucradas. Esencialmente la geometría epipolar caracteriza la relación entre un par de cámaras como la intersección de sus planos imagen con una familia de planos que contengan la línea que une los centros de cámaras (línea base o *baseline*). El estudio de esta relación es de interés dado que permite restringir el espacio de búsqueda al momento de asociar puntos entre los planos imagen.

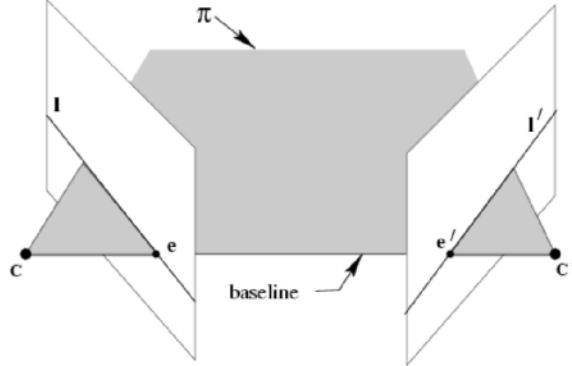


Fig. 3.2.1: La línea base interseca con cada plano imagen en los epipolos e y e' . Todo plano π que contenga la línea base es un plano epipolar, e interseca los planos imagen formando las líneas epipoles l y l' .

3.2.1. Correspondencia entre puntos

Sea X un punto en el espacio 3D observado simultáneamente por dos cámaras distintas C y C' , de manera que x representa la proyección de X en el plano imagen de C y x' la proyección de X en el plano imagen de C' . Es posible observar que los puntos imagen x y x' , el punto en el espacio X y los centros de cámara son coplanares (Fig.(a) 3.2.2). Es decir, pertenecen a un mismo plano al cual denominaremos como π . Más aún, las rectas formadas por el centro de cada cámara y los puntos en la imagen x y x' intersecan en el punto del espacio 3D X .

Suponiendo el escenario donde se conozca solamente el punto x perteneciente al plano imagen de la cámara C , se puede restringir el espacio donde buscar el correspondiente punto x' en el plano imagen de la cámara C' . El plano π puede ser determinado por medio de la línea base entre los centros de cámara y el vector dirección que une el centro de la cámara C con el punto imagen x . De esta forma, sabiendo que x' debe pertenecer a π , se asegura que x' debe pertenecer a la recta l' definida por la intersección entre el plano π y el plano imagen de la cámara C' (Fig.(b) 3.2.2). La recta l' se conoce como la línea epipolar (*epipolar line*) de x y es especialmente útil para acotar la búsqueda del correspondiente punto imagen x' en algoritmos de asociación de características visuales.

3.2.2. Rectificación estéreo

La rectificación estéreo proyecta las imágenes obtenidas por las cámaras a un plano imagen en común, de manera que puntos correspondientes (3.2.1) se encontrarán alineados en una misma fila (Fig. 3.2.3). Esta proyección permite considerar las cámaras involucradas como paralelas, es decir, relacionadas por únicamente por una translación en el eje horizontal la cual se conoce como línea base (*baseline*).

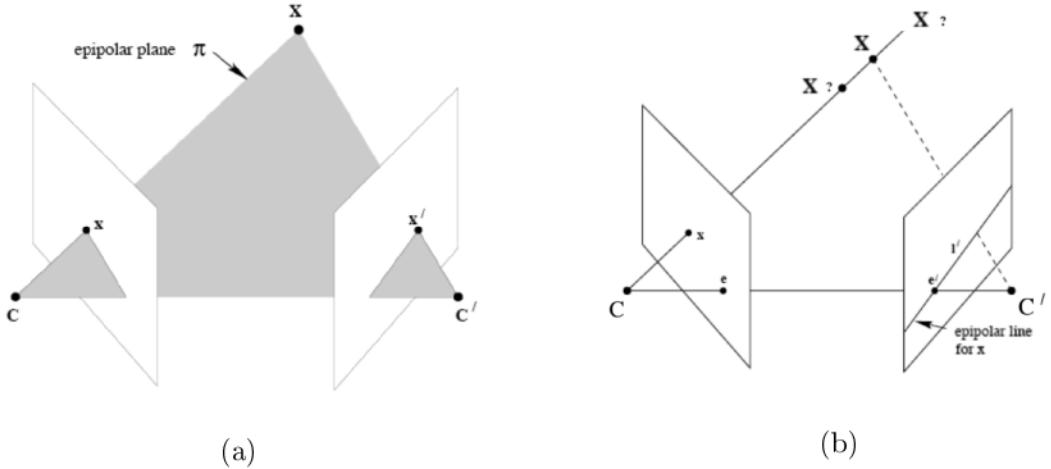


Fig. 3.2.2: (a) Los centros de cámara, el punto 3D X y los puntos imagen x y x' pertenecen a un mismo plano π . (b) La recta definida por el centro de cámara C y el punto imagen x es observada en el plano imagen de C' como la línea epipolar l' . El punto en el espacio X , el cual es proyectado en el plano imagen de C como x , debe ser observado por la cámara C' sobre la línea epipolar l' .

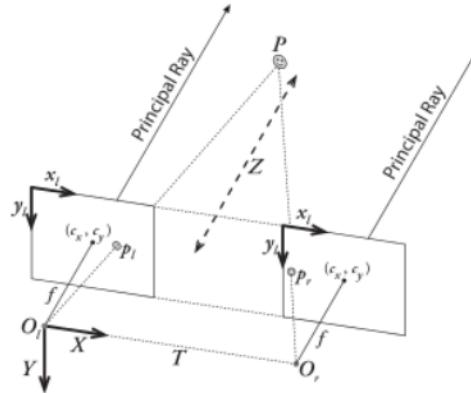


Fig. 3.2.3: Par estéreo rectificado, notar que los planos imagen de las cámaras se encuentran alineadas

Luego de la rectificación, la búsqueda de correspondencias entre las imágenes es simplificada a una búsqueda horizontal sobre la misma fila. Este resultado es interesante para el cálculo de la disparidad entre características visuales, la cual permite estimar la profundidad (posición en el espacio) con respecto a la cámara en que se encuentran las marcas del ambiente observadas.

3.2.3. Triangulación de puntos del espacio

Dada una rectificación estéreo se puede realizar una reconstrucción 3D de los puntos presentes en las imágenes. La Figura (3.2.4) muestra la geometría involucrada durante la triangulación de un punto P del espacio. Siendo $P = (X, Y, Z)$ un punto 3D del espacio, (x^l, y^l) y (x^r, y^r) su proyección en la cámara izquierda y derecha respectivamente y T la línea base que une las cámaras, las coordenadas de P pueden ser derivadas a través de propiedades propias de triángulos semejantes. Específicamente, las siguientes relaciones pueden ser derivadas:

$$X = \frac{x^l \cdot Z}{f} \quad Y = \frac{y^l \cdot Z}{f} \quad Z = \frac{T \cdot f}{x^l - x^r} \quad (3.2.1)$$

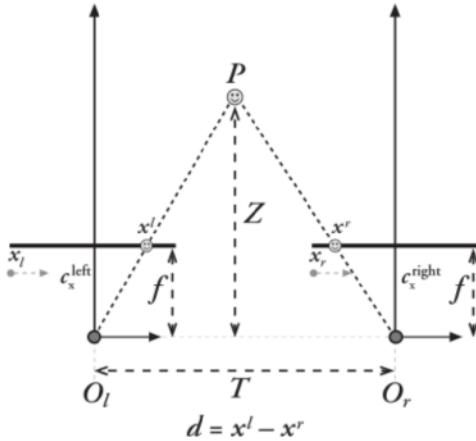


Fig. 3.2.4: Escenario de triangulación estéreo con cámaras rectificadas donde se desea calcular las coordenadas de P , en especial la profundidad Z . Los parámetros intrínsecos como la distancia focal f y puntos principales $c^{\text{left}}, c^{\text{right}}$ son conocidos, así como también los centros de cámara O_l, O_r y la línea base T .

Si bien la triangulación requiere de formulas sencillas es interesante observar que la capacidad de observar la profundidad en que se encuentra un punto del espacio esta condicionada por la resolución de las cámaras. Se define la disparidad como la distancia existente entre las proyecciones de las diferentes cámaras como $d = x^l - x^r$ y es interesante notar que la profundidad Z es inversamente proporcional a la disparidad d (3.2.1). Puntos del ambiente cercanos presentaran mayor disparidad en las proyecciones que puntos distantes. Cuando el valor de d es cercano a 0, pequeñas diferencias de disparidad producen un gran cambio en la profundidad percibida y cuando el valor de d es grande, pequeñas diferencias de disparidad producen pequeños cambios en la profundidad. La consecuencia de esto es que la reconstrucción 3D del ambiente utilizando cámaras estéreo es más precisa para puntos cercanos a la cámara.

Capítulo 4

Método propuesto de mapeo denso basado en disparidad

A continuación se expone en detalle la solución propuesta para el problema de mapeo denso basado en disparidad. El método fue concebido para ejecutarse en paralelo y conjuntamente al sistema S-PTAM (*Stereo Parallel Tracking and Mapping*), de forma que la arquitectura y las diferentes técnicas involucradas en el desarrollo están motivadas por las características propias del sistema de SLAM estéreo considerado.

4.1. Esquema general

El sistema de mapeo denso se implementó como un nodo del framework ROS, íntegramente desacoplado de S-PTAM. Notar que esta generalidad en la implementación permite fácilmente re-utilizar el nodo de densificación con otros métodos de SLAM que presenten una interfaz similar.

Los *keyframes* procesados por S-PTAM son encolados para ser tratados en el hilo de *Disparity map thread* y posteriormente en el hilo de *3D projection thread* (Fig. 4.1.1).

Disparity map thread se encarga de procesar el par de imágenes estéreo de cada *keyframe* de forma secuencial, computando un mapa bidimensional donde a cada píxel se le asignada un valor de disparidad. El mapa resultante es encolado para su posterior tratamiento en el hilo de *3D projection thread*.

Una vez que S-PTAM computó la pose para el *keyframe* en cuestión, *3D projection thread* utiliza esta estimación para ajustar y filtrar el mapa mediante un chequeo de consistencia y redundancia, que consiste en la proyección de la nube de puntos global sobre el *keyframe* actual. Finalmente, el mapa resultante es re-proyectado al espacio tridimensional y añadido al mapa global.

Un tercer hilo *Refinement thread* es ejecutado con menor prioridad para ajustar la nube de puntos a medida que S-PTAM publica poses refinadas. A su vez, este hilo se encarga de realizar un swap de nubes de puntos a memoria secundaria, necesario en datasets de gran escala.

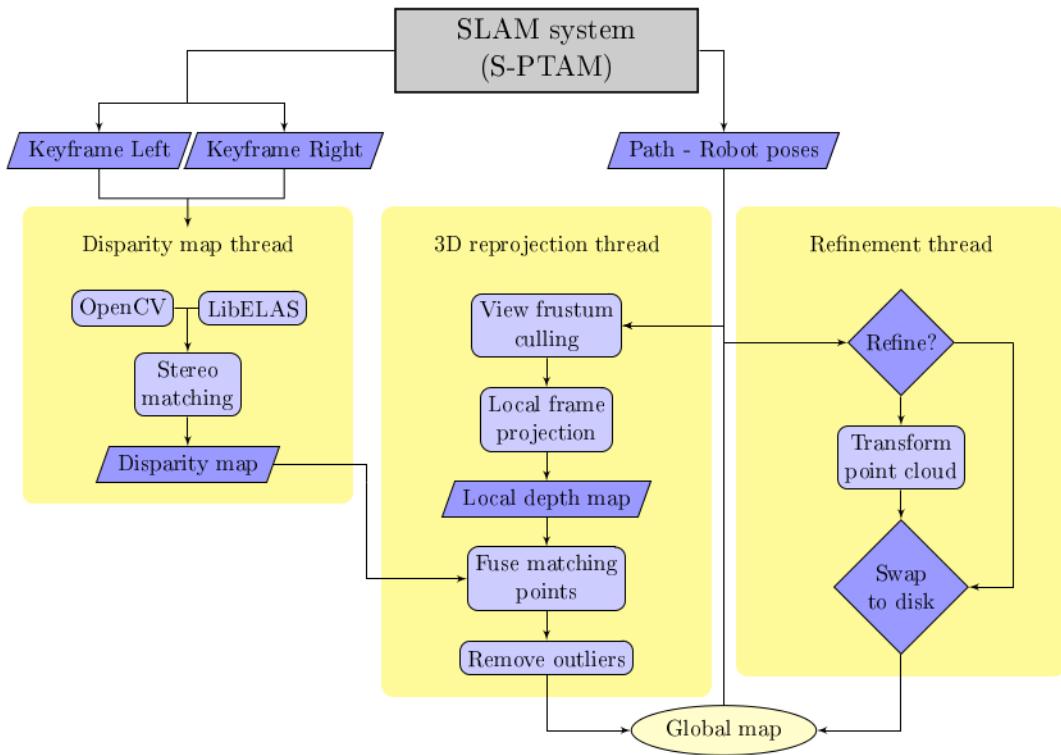


Fig. 4.1.1: Esquema general del sistema de densificación de mapas en tiempo real, procesando la salida del sistema de SLAM (e.g. S-PTAM).

4.2. Cálculo de mapas de disparidad

Para la obtención de mapas de disparidad se utilizaron dos métodos diferentes, que cuentan con implementaciones de código abierto disponibles públicamente. En ambos casos, se analizó el desempeño y los resultados obtenidos, para determinar la adaptación del método a los requerimientos del sistema.

4.2.1. OpenCV

OpenCV es una biblioteca libre de visión artificial originalmente desarrollada por Intel, que provee un método de correspondencia estereópica llamado “Block matching (BM) algorithm”.

Este algoritmo, similar al algoritmo desarrollado por Kurt Konolige [24], funciona utilizando pequeñas ventanas de “sumas de diferencia absoluta” (SAD) para encontrar puntos coincidentes entre el par de imágenes estereopares rectificadas.

Si bien este método es muy rápido y efectivo, solo encuentra aquellos puntos que presentan alta textura en ambas imágenes, por lo que puede resultar inadecuado para entornos complejos como puede distinguirse en 4.2.1 y 4.2.2.

4.2.2. LIBELAS

LIBELAS (Library for Efficient Large-scale Stereo Matching) [25] es una librería multiplataforma de código abierto para computar mapas de disparidad a partir de pares de imágenes estereopares rectificadas de alta resolución.

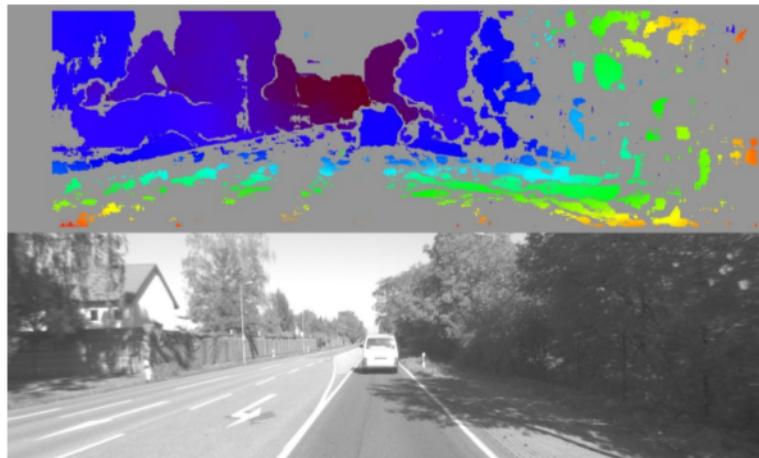


Fig. 4.2.1: Arriba: Mapa de disparidad calculado mediante OpenCV StereoBM. Tiempo de ejecución: 0.019s. Abajo: Imagen izquierda perteneciente al par estéreo 27, del dataset KITTI secuencia 04.

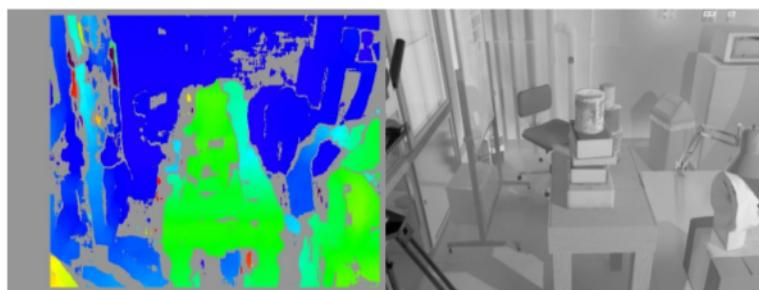


Fig. 4.2.2: Arriba: Mapa de disparidad calculado mediante OpenCV StereoBM. Tiempo de ejecución: 0.011s. Abajo: Imagen izquierda perteneciente al par estéreo 221, del dataset Tsukuba Day-light.

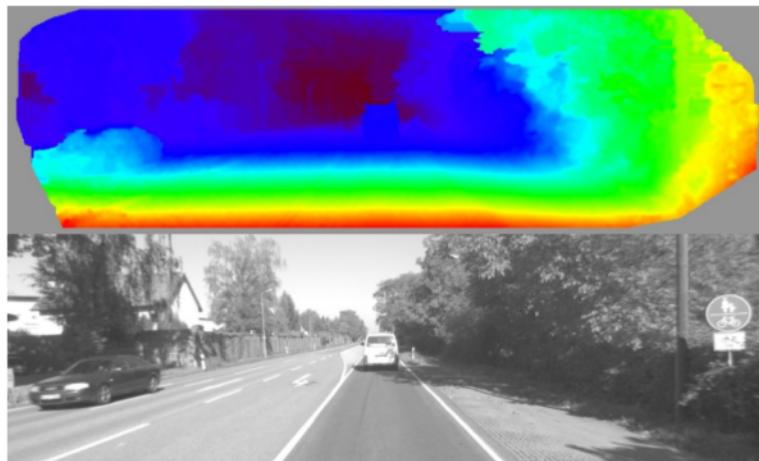


Fig. 4.2.3: Arriba: Mapa de disparidad calculado mediante LibELAS. Tiempo de ejecución: 0.164s.
Abajo: Imagen izquierda perteneciente al par estéreo 22, del dataset KITTI secuencia 04.

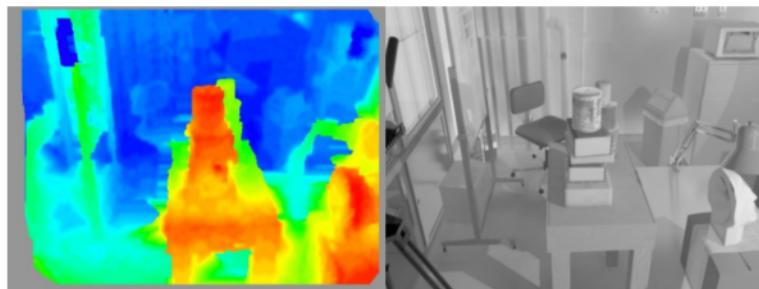


Fig. 4.2.4: Arriba: Mapa de disparidad calculado mediante LibELAS. Tiempo de ejecución: 0.151s.
Abajo: Imagen izquierda perteneciente al par estéreo 222, del dataset Tsukuba Daylight.

El método empleado se basa en que, pese al hecho de que muchas correspondencias estéreo son altamente ambiguas, algunas de ellas pueden ser robustamente matcheadas. Asumiendo variaciones suaves en la disparidad es posible sectorizar la imagen, mediante la triangulación de Delaunay, sobre estos 'puntos de soporte' para la estimación de las restantes disparidades. En [4.2.3](#) y [4.2.4](#) pueden observarse resultados de su ejecución.

4.3. Proyección al plano de la imagen

4.3.1. Frustum of view

El view frustum es la región 3D que contiene los puntos que son potencialmente (pueden existir occlusiones) visibles en el plano de la imagen dada una determinada posición de la cámara. Esta región es determinada por los parámetros de la cámara y, en las cámaras estenopeicas (pinhole cameras), tiene la forma de una pirámide truncada.

El vértice de la pirámide es la posición de la cámara y su base está definida por el plano lejano -far plane- potencialmente infinito. A su vez, la pirámide es truncada por un plano cercano -near plane- que da el nombre de Frustum of view. Todo lo que es potencialmente proyectable sobre el plano de la imagen se encuentra dentro de la pirámide truncada, por lo que el resto de los puntos pueden descartarse sin necesidad de realizar la proyección.

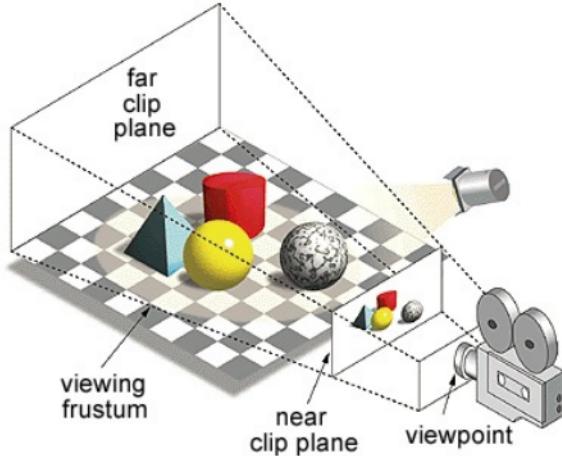


Fig. 4.3.1: Representación de la pirámide trunca “Frustum of view”. Todo lo que se encuentra entre los planos cercano y lejano es el área de visión de la cámara.

Para el keyframe K_i , se calcula su entorno local V_i como la unión de las últimas N nubes de puntos, donde P_j es la nube de puntos generada desde el keyframe K_j .

$$V_i = \{P_{i-N}, \dots, P_{i-1}\}$$

El Frustum of view es utilizado para filtrar la nube de puntos del entorno local V_i .

Sean $\pi_{i,[1,\dots,6]}$ los planos correspondientes al Frustum of view del keyframe K_i , expresados en su forma general como:

$$\pi_{i,k} = (a_{i,k}, b_{i,k}, c_{i,k}, d_{i,k}) \text{ tal que}$$

$$a_{i,k} \cdot x + b_{i,k} \cdot y + c_{i,k} \cdot z + d_{i,k} \cdot w = 0 \text{ para cada } (x, y, z, w) \in \pi_{i,k}$$

Notar que los puntos del espacio 3D se encuentran representados en sus coordenadas homogéneas. Para más detalle sobre la obtención de los planos ver [apéndice sobre frustum].

Asumiendo que los planos tienen su vector normal apuntando hacia el interior de la pirámide, el entorno local filtrado V'_i queda determinado por:

$$V'_i = \{p \in V_i \text{ tal que } p \cdot \pi_{i,k} \geq 0 \text{ para cada } k \in [1, \dots, 6]\}$$

4.3.2. Proyección al plano de la imagen - Obtención del depth map

Como se vio en la sección X, dada la posición del keyframe puede obtenerse la matriz de proyección P para proyectar puntos en el sistema de coordenadas del mundo sobre el plano de la imagen.

La nube del entorno local, filtrada mediante el Frustum of View, es proyectada punto por punto sobre el keyframe actual. Si dos puntos del espacio corresponden al mismo píxel, el más lejano es descartado, dado que se ve ocluido por el más cercano y por lo tanto no es visible desde la perspectiva de la cámara.

La matriz resultante asigna a cada píxel un valor numérico que representa la profundidad en Z desde la perspectiva de la cámara. Esto se conoce como mapa de profundidad -depth map-.

TODO: pseudocódigo de proyección.

4.4. Consistencia y redundancia (StereoScan)

Cada punto está asociado al keyframe original del que fueron triangulados y se le asigna un contador de la veces que fue triangulado desde distintos keyframes.

4.4.1. Heurística de detección de outliers y fusión.

Basado en el trabajo del paper Stereoscan, se utilizan ambos mapas de profundidad obtenidos en las secciones (disparidad) y (proyección al plano de la imagen).

Aquellos píxeles que poseen una profundidad válida en ambos mapas y cuya diferencia es menos que un cierto threshold ξ se fusionan. El punto no se agrega nuevamente al mapa denso global, sino que permanece asociado solamente a la nube del keyframe que lo trianguló originalmente, y su profundidad en Z se ajusta a la media aritmética de ambos valores.

Si la profundidad en ambos mapas -(disparidad) y (proyección al plano de la imagen)- difiere en un valor mayor al threshold ξ , un nuevo punto se triangula desde el keyframe actual con un valid_counter de 1. En consecuencia, el punto previo (triangulado en un keyframe anterior) disminuye su valid_counter en 1, siendo descartado cuando este se vuelve nulo.

Utilizando el método descripto, aquellos puntos que presentan inconsistencias son rápidamente descartados, reduciendo drásticamente el número de outliers . De la misma manera, la nube es refinada constantemente, fusionando en Z los puntos existentes con la nueva disparidad triangulada en cada nuevo keyframe.

TODO: mostrar números sobre la reducción de complejidad del mapa.

TODO: mostrar la aceleración en la computación de cada keyframe por StereoScan.

4.5. Detalles de implementación

- Hilos, colas, locks
- Swap a disco

Capítulo 5

Experimentación y resultados

Capítulo 6

Conclusiones

6.1. Trabajo futuro

Bibliografía

- [1] Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Computer vision and pattern recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 519–528. IEEE, 2006.
- [2] Reinhard Koch, Marc Pollefeys, and Luc Van Gool. Multi viewpoint stereo from uncalibrated video sequences. In *European conference on computer vision*, pages 55–71. Springer, 1998.
- [3] Marc Pollefeys, Luc Van Gool, Maarten Vergauwen, Frank Verbiest, Kurt Cornelis, Jan Tops, and Reinhard Koch. Visual modeling with a hand-held camera. *International Journal of Computer Vision*, 59(3):207–232, 2004.
- [4] Lindsay Kleeman. Advanced sonar and odometry error modeling for simultaneous localisation and map building. In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 1, pages 699–704. IEEE, 2003.
- [5] Fabrizio Abrate, Basilio Bona, and Marina Indri. Experimental ekf-based slam for mini-rovers with ir sensors only. In *EMCR*, 2007.
- [6] David M Cole and Paul M Newman. Using laser range data for 3d slam in outdoor environments. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 1556–1563. IEEE, 2006.
- [7] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 13(2):99–110, 2006.
- [8] Tim Bailey and Hugh Durrant-Whyte. Simultaneous localization and mapping (slam): Part ii. *IEEE Robotics & Automation Magazine*, 13(3):108–117, 2006.
- [9] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [10] Emanuele Trucco and Alessandro Verri. *Introductory techniques for 3-D computer vision*, volume 201. Prentice Hall Englewood Cliffs, 1998.
- [11] Yi Ma, Stefano Soatto, Jana Kosecka, and S Shankar Sastry. *An invitation to 3-d vision: from images to geometric models*, volume 26. Springer Science & Business Media, 2012.

- [12] Hao Li, Bart Adams, Leonidas J Guibas, and Mark Pauly. Robust single-view geometry and motion reconstruction. In *ACM Transactions on Graphics (TOG)*, volume 28, page 175. ACM, 2009.
- [13] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11):1330–1334, 2000.
- [14] Paul Beardsley, Phil Torr, and Andrew Zisserman. 3d model acquisition from extended image sequences. In *European conference on computer vision*, pages 683–695. Springer, 1996.
- [15] Andrew W Fitzgibbon and Andrew Zisserman. Automatic camera recovery for closed or open image sequences. In *European conference on computer vision*, pages 311–326. Springer, 1998.
- [16] Marc Pollefeys, Reinhard Koch, and Luc Van Gool. Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters. *International Journal of Computer Vision*, 32(1):7–25, 1999.
- [17] Frédéric Devernay and Olivier D Faugeras. Automatic calibration and removal of distortion from scenes of structured environments. In *SPIE’s 1995 International Symposium on Optical Science, Engineering, and Instrumentation*, pages 62–72. International Society for Optics and Photonics, 1995.
- [18] Nico Cornelis, Bastian Leibe, Kurt Cornelis, and Luc Van Gool. 3d urban scene modeling integrating recognition and reconstruction. *International Journal of Computer Vision*, 78(2-3):121–141, 2008.
- [19] Marc Pollefeys, David Nistér, J-M Frahm, Amir Akbarzadeh, Philipppos Mordohai, Brian Clipp, Chris Engels, David Gallup, S-J Kim, Paul Merrell, et al. Detailed real-time urban 3d reconstruction from video. *International Journal of Computer Vision*, 78(2-3):143–167, 2008.
- [20] Taihú Pire, Thomas Fischer, Javier Civera, Pablo De Cristóforis, and Julio Jacobo Berlles. Stereo parallel tracking and mapping for robot localization. In *International Conference on Intelligent Robots and Systems*, Hamburg, Germany, September 2015.
- [21] Georg Klein and David Murray. Parallel Tracking and Mapping for Small AR Workspaces. In *ISMAR*, pages 1–10, Washington, DC, USA, 2007. IEEE Computer Society.
- [22] Jangheon Kim and Thomas Sikora. Gaussian scale-space dense disparity estimation with anisotropic disparity-field diffusion. In *3-D Digital Imaging and Modeling, 2005. 3DIM 2005. Fifth International Conference on*, pages 556–563. IEEE, 2005.
- [23] Christoph Strecha and Luc Van Gool. Pde-based multi-view depth estimation. In *3D Data Processing Visualization and Transmission, 2002. Proceedings. First International Symposium on*, pages 416–425. IEEE, 2002.
- [24] Kurt Konolige. Small vision systems: Hardware and implementation. In *Robotics research*, pages 203–212. Springer, 1998.

- [25] Andreas Geiger, Julius Ziegler, and Christoph Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 963–968. IEEE, 2011.
- [26] Andreas Geiger, Martin Roser, and Raquel Urtasun. Efficient large-scale stereo matching. In *Asian conference on computer vision*, pages 25–38. Springer, 2010.
- [27] Jan Stühmer, Stefan Gumhold, and Daniel Cremers. Real-time dense geometry from a handheld camera. In *Joint Pattern Recognition Symposium*, pages 11–20. Springer, 2010.
- [28] Gottfried Graber, Thomas Pock, and Horst Bischof. Online 3d reconstruction using convex optimization. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 708–711. IEEE, 2011.
- [29] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *2011 international conference on computer vision*, pages 2320–2327. IEEE, 2011.
- [30] Raul Mur-Artal and Juan D Tardós. Probabilistic semi-dense mapping from highly accurate feature-based monocular slam. In *Robotics: Science and Systems*, 2015.
- [31] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *European Conference on Computer Vision*, pages 834–849. Springer, 2014.
- [32] Jakob Engel, Jörg Stückler, and Daniel Cremers. Large-scale direct slam with stereo cameras. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 1935–1942. IEEE, 2015.
- [33] David Caruso, Jakob Engel, and Daniel Cremers. Large-scale direct slam for omnidirectional cameras. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 141–148. IEEE, 2015.
- [34] Alejo Concha, Giuseppe Loianno, Vijay Kumar, and Javier Civera. Visual-inertial direct slam. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 1331–1338. IEEE, 2016.
- [35] Alejo Concha and Javier Civera. Using superpixels in monocular slam. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 365–372. IEEE, 2014.
- [36] Alejo Concha and Javier Civera. Dpptam: Dense piecewise planar tracking and mapping from a monocular sequence. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 5686–5693. IEEE, 2015.
- [37] Javier Civera, Dorian Gálvez-López, Luis Riazuelo, Juan D Tardós, and JMM Montiel. Towards semantic slam using a monocular camera. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 1277–1284. IEEE, 2011.
- [38] Renato F Salas-Moreno, Richard A Newcombe, Hauke Strasdat, Paul HJ Kelly, and Andrew J Davison. Slam++: Simultaneous localisation and mapping at the level of objects.

In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1352–1359, 2013.

- [39] Varsha Hedau, Derek Hoiem, and David Forsyth. Recovering the spatial layout of cluttered rooms. In *Computer vision, 2009 IEEE 12th international conference on*, pages 1849–1856. IEEE, 2009.
- [40] Alex Flint, David Murray, and Ian Reid. Manhattan scene understanding using monocular, stereo, and 3d features. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2228–2235. IEEE, 2011.
- [41] Sid Yingze Bao, Manmohan Chandraker, Yuanqing Lin, and Silvio Savarese. Dense object reconstruction with semantic priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1264–1271, 2013.
- [42] Vibhav Vineet, Ondrej Miksik, Morten Lidegaard, Matthias Nießner, Stuart Golodetz, Victor A Prisacariu, Olaf Kähler, David W Murray, Shahram Izadi, Patrick Pérez, et al. Incremental dense semantic stereo fusion for large-scale semantic scene reconstruction. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 75–82. IEEE, 2015.
- [43] Veeravalli Seshadri Varadarajan. *Lie groups, Lie algebras, and their representations*, volume 102. Springer Science & Business Media, 2013.