



UNIVERSIDAD DE GUANAJUATO

DIVISIÓN DE INGENIERÍAS CAMPUS
IRAPUATO-SALAMANCA

*“Diseño del un algoritmo de realidad
aumentada para la exploración de un entorno
semi-estructurado”*

TESIS PROFESIONAL

QUE PARA OBTENER EL GRADO DE:
LICENCIATURA EN INGENIERÍA ELECTRÓNICA

PRESENTA:

Liliana Caritina Orozco Pérez

ASESORES:

Dr. Juan Gabriel Aviña Cervantes

M.I. Pedro Alfonso Patlán Rosales

Agradecimientos Institucionales

- A la **Universidad de Guanajuato**, por el apoyo que me brindó por medio de la rectoría del Campus Irapuato-Salamanca y la dirección de la División de Ingenierías para poder realizar mi estancia de investigación en el INRIA de Sophia Antipolis, Francia.



- Al **Consejo Nacional de Ciencia y Tecnología (CONACyT)**, por el apoyo que me brindó al otorgarme la beca para posgrado con el número de apoyo 242120.



- Al **INRIA-Sophia Antipolis**, por el apoyo durante mi estancia de investigación en el equipo EVOLUTION.



Agradecimientos

A Dios: Por ayudarme a terminar este proyecto dandome la fortaleza para continuar mi superación aún cuando existán momentos complicados.

A mis padres: A mi padre, quiero agradecer todo su apoyo porque como siempre me animó a seguir adelante; sus principios y sus experiencias me hacen cada día dar lo mejor de mí. A mi madre, porque nunca me deja solo, siempre me escucha y orienta en todo momento de la mejor manera posible. Por último, quiero decirles a los dos que siempre serán mis ídolos y que este trabajo de tesis es dedicado a ustedes por todo el esfuerzo que han hecho por mí.

Abstract

*This thesis describes a dense **visual odometry** and **3D reconstruction system**. The system makes no prior assumptions about the camera motion, and it uses a disparity map obtained from successive stereo image pairs to estimate the camera motion and the 3D reconstruction of the explored environment. The system has been tested with virtual environments and real indoor environments. Results are visualized in a graphical interface which was developed during this project.*

Resumen

En este trabajo se describe un sistema para reconstruir una escena tridimensional y encontrar la odometría de un par de cámaras usando solo la información proveniente de éstas. El sistema utiliza el mapa de disparidades que se obtiene de cada par de imágenes en una posición de referencia para encontrar la profundidad por medio de los parámetros intrínsecos de la cámara. Una vez que se obtiene la profundidad en la referencia se compara con el mapa de disparidades del siguiente par de imágenes con el objetivo de que el sistema estime el movimiento de la cámara, y al mismo tiempo construir la escena en tres dimensiones. Este sistema se ha probado usando entornos virtuales y entornos de interior reales. Los resultados se visualizan en una interfaz gráfica, la cual se desarrolló para este proyecto.

Índice general

1. Introducción	1
1.1. Estado del Arte	2
1.2. Motivaciones y Aportación	6
1.3. Objetivos	6
1.4. Organización del Documento	7
2. Entornos de Navegación	9
2.1. Entorno Virtual	9
3. Estimación de la Odometría y Reconstrucción 3D	10
3.1. Mapa de Profundidad	10
4. Visualización de la Odometría y Reconstrucción 3D	12
4.1. Estructura de la Interfaz Gráfica	12
5. Conclusiones	14
5.1. Trabajo a Futuro	15
A. Jacobiano Actual y Jacobiano de Referencia	17
A.1. Jacobiano Actual	17
A.2. Jacobiano de Referencia	20
Bibliografía.	24

Índice de figuras

1.1. Robot Stanford Cart.	3
1.2. Mapa denso y SLAM 3D Lemaire	5
1.3. Reconstrucción 3D Geiger 2011	6
1.4. Sistema esencial del proyecto.	7

Índice de tablas

CAPÍTULO 1

Introducción

Contenido

1.1. Estado del Arte	2
1.2. Motivaciones y Aportación	6
1.3. Objetivos	6
1.4. Organización del Documento	7

La navegación autónoma de un robot móvil ha sido uno de los temas más explorados por la comunidad científica en todo el mundo. La principal razón por la que surge esta línea de investigación se debe a la necesidad de explorar lugares que son de alto riesgo para un ser vivo o lugares pequeños donde el ser humano no puede entrar. Estas razones han llevado a investigadores a pensar en sistemas electromecánicos que puedan tener facilidad de movimientos y percepción. Estos sistemas, llamados *robots*, iniciaron una revolución que se refleja en vastas investigaciones y desarrollos para lograr su autonomía. Para lograr la percepción del robot se inició con el desarrollo de dispositivos electrónicos que convierten una sensación en señales que puedan caracterizarse, estos dispositivos son llamados *sensores*. En la actualidad existe una amplia variedad de sensores basados en el avance tecnológico para lograr una buena percepción.

Los sensores se han utilizado para lograr la interacción del robot y el ambiente, no obstante las señales adquiridas por los sensores requieren ser procesadas de tal manera que el robot pueda tomar decisiones basadas en éstas. Generalmente la decisión no solo se basa en un filtrado de la señal, sino en algoritmos inteligentes. Los movimientos del robot no solo se basan en sensores que perciben el ambiente, también se consideran los factores inherentes al robot, como es el deslizamiento de las

ruedas y el tipo de superficie por el cual se mueve. Estos últimos factores pueden ser corregidos tomando un muestreo de las señales para estimar la posición y orientación del robot con base en un estado inicial. La estimación de la posición y orientación se le conoce como *odometría* y se calcula con base a un sensor o la conjunción de dos o más sensores. Por ejemplo, el dispositivo *kinect* de Microsoft® está compuesto por una cámara a color y dos sensores de profundidad que usan un láser infrarrojo, logrando una reconstrucción 3D del área de visión.

En este proyecto se pretende realizar el cálculo de la odometría de un robot basándose en un sensor muy completo, el sensor visual. Éste ofrece la ventaja de captar una gran cantidad de información del entorno, similar al sistema de visión humano. La información que proporciona este sensor es necesaria para estimar el movimiento del robot. En este proyecto, además de buscar calcular la odometría en una trayectoria por la cual se desplaza el robot, se obtendrá la reconstrucción 3D del entorno que se observa durante dicha trayectoria. La reconstrucción de este entorno es posible gracias a la visión estereoscópica, logrando un mapeo 3D el cual también se describe a detalle en esta tesis. En la siguiente sección se presentan trabajos relacionados con este proyecto.

1.1. Estado del Arte

Las referencias relacionadas con la odometría visual comienzan desde la década de los setentas, cuando se desarrolló el robot Hilare [Giralt et al., 1979] en el LAAS (*Laboratoire d'Analyse et d'Architecture des Systèmes*). En él, se propuso un sistema para estimar la trayectoria del robot sensando obstáculos con forma de poliedros. Para encontrar la trayectoria óptima y llegar a un objetivo, se usó la proyección de los obstáculos en el piso como referencia. El sistema del Hilare se compone de un sistema robótico diferencial que con la ayuda de una cámara y un láser (*range finder*) logran la percepción de un entorno estructurado. En el mismo año de esta aportación, se presentó un trabajo para lograr la construcción de un mapa de un entorno por el cual se mueve un robot en configuración Ackerman, como se observa

en la Figura 1.1. El robot tiene la ventaja de que solo usa una cámara para lograr construir el mapa de su entorno. La publicación de este trabajo lleva por nombre *Visual Mapping by a Robot Rover* [Moravec, 1979]. Este robot usa un sistema de una cámara montada sobre un mecanismo que ayuda a que ésta se pueda mover de manera horizontal en diferentes posiciones con un paso de deslizamiento horizontal de 6.4 cm, lo cual hace posible que se obtengan varias imágenes del mismo entorno. Éste cambio en la posición de la cámara fue de gran ayuda para lograr la localización de características en las imágenes por medio de la correlación de cada secuencia de imágenes, logrando así el principio de un sistema binocular calibrado para encontrar la profundidad. El cumplimiento de los objetivos es lento debido a los cambios de posición en la cámara, pero ha sido la base para el desarrollo de trabajos posteriores.



Figura 1.1: Robot Stanford Cart.

Hasta ese momento solo se habían desarrollado sistemas monoculares, poco tiempo después comenzó a experimentar con sistemas binoculares. Matthies comenzó a usar este tipo de sistemas para lograr la navegación de un robot. En uno de sus primeros trabajos [Matthies and Shafer, 1987] se presenta la metodología de la navegación basada en un par estéreo en un entorno de interior; el trabajo presenta resultados que logran una precisión del 2% en una distancia de 5.5 m con 55 pares de imágenes. Matthies en su tesis de doctorado titulada *Dynamic Stereo Vision* [Matthies, 1989] presentó un sistema binocular montado sobre un robot para estimar el movimiento del mismo basado en el rastreo de puntos de interés en 3D (*landmarks*). En este trabajo de Matthies se define la tarea de estimación del movimiento que realiza el

móvil como un problema estadístico. También se implementó un algoritmo para la relación de intensidades basado en un modelo Bayesiano con el fin de obtener la información de la profundidad en cada secuencia de imágenes.

Los trabajos sobre la odometría visual fueron avanzando conforme a la necesidad de ser más precisos en el cálculo de la estimación de la trayectoria. Uno de los problemas que requiere de gran precisión se encuentra en los robots para la exploración espacial. Matthies comenzó a hacer pruebas de odometría visual en este ámbito con el prototipo llamado Rocky 7 [Olson and Matthies, 1998] en el cual se formuló la localización como un problema estadístico. Este trabajo presenta una solución basada en la máxima verosimilitud de un sistema estéreo, generando resultados con gran exactitud en la localización. Este trabajo se extendió en la publicación de [Olson et al., 2003] donde se presenta el algoritmo para la navegación usando *ego-motion* (odometría visual) que fue diseñado para modelar el error de la relación estéreo como una distribución normal. Esto ayudó a que el error en la posición no afectara en la precisión de la distancia recorrida por el robot explorador.

En la navegación también se desarrolló la odometría visual, de tal manera que se implementó un sistema binocular para lograr la “Localización y Mapeo Simultáneo” (SLAM, en Inglés) en dos y tres dimensiones. Una publicación que presenta lo antes descrito se encuentra en [Nistér et al., 2004] donde se compara la precisión de la odometría visual con dos tipos de *Sistemas de Posicionamiento Global (GPS)*, diferencial e inercial, presentando resultados cuantitativamente muy buenos en comparación a estos sistemas.

Las evidencias de que la odometría visual es de gran ayuda para la localización y mapeo de un robot se observan en aportaciones recientes. Para el SLAM 3D en [Lemaire et al., 2007] se presentan dos enfoques que abordan el problema usando visión artificial: visión estéreo e imágenes de una sola cámara. Ambos enfoques utilizan un algoritmo robusto para la relación de puntos de interés que trabaja en entornos de interior y exterior. El enfoque basado en la estéreo visión es una implementación clásica del SLAM y el otro enfoque monocular presenta una nueva manera de buscar

puntos de interés. Se analizan los dos métodos propuestos y se presentan resultados con un robot explorador y con un pequeño dirigible (ver Figura 1.2).

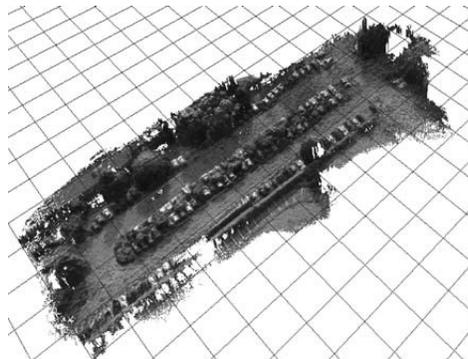


Figura 1.2: Mapa con estéreo visión densa y estéreo SLAM para la trayectoria (cuadros de 10 m) [Lemaire et al., 2007].

En años más recientes, [Geiger et al., 2011] han propuesto un nuevo enfoque para la reconstrucción 3D de mapas en tiempo real, usando secuencias de imágenes en alta resolución provenientes de un par estéreo. Este algoritmo se apoya en recientes avances de la relación estéreo y una odometría visual basada en características (esquinas y puntos de interés). La generación de la reconstrucción 3D en esta publicación (ver Figura 1.3) tiene un desempeño en su odometría de 25 fps y el mapa de profundidad a una razón de 3-4 fps. Este tipo de sistemas también tiene la ventaja de la velocidad de los microprocesadores actuales y la gran potencia de la computadoras.

Actualmente se siguen ampliando los horizontes de la velocidad en la reconstrucción 3D con el uso de la visión al utilizar la tecnología de unidades de procesamiento gráfico de alto rendimiento, como la Arquitectura de Dispositivos de Cómputo Unificado (CUDA, por sus siglas en Inglés) que contiene un conjunto de instrucciones en C desarrolladas por NVIDIA[®]. Esto sin duda hará que se tenga un gran avance en cuanto a velocidad de procesamiento los algoritmos de visión artificial en un futuro no muy lejano.



Figura 1.3: Resultados de la reconstrucción 3D en tiempo real [Geiger et al., 2011].

1.2. Motivaciones y Aportación

Las aportaciones que proveen los trabajos citados en la sección anterior son solo algunos de los muchos trabajos relacionados con la odometría visual y la reconstrucción 3D. En la actualidad, uno de los aspectos que se desea mejorar es la rapidez en el cálculo de la odometría visual. En este proyecto se propone un sistema eficiente para la estimación de la odometría visual, tomando la mayor cantidad de información visual posible proveniente de un par estéreo y, mediante esta información visual, realizar la reconstrucción 3D del entorno del sistema. Otra propuesta para este proyecto es la creación de entornos virtuales para probar la eficiencia y eficacia del sistema mencionado. Además, también se presenta una interfaz gráfica multiplataforma para la visualización e interacción con los resultados de la odometría visual y reconstrucción 3D.

1.3. Objetivos

Para lograr el sistema planteado, se establecieron los siguientes objetivos:

- Adquisición de las imágenes de un entorno.
- Obtención de la odometría visual.

- Reconstrucción 3D del entorno.
- Análisis de resultados.

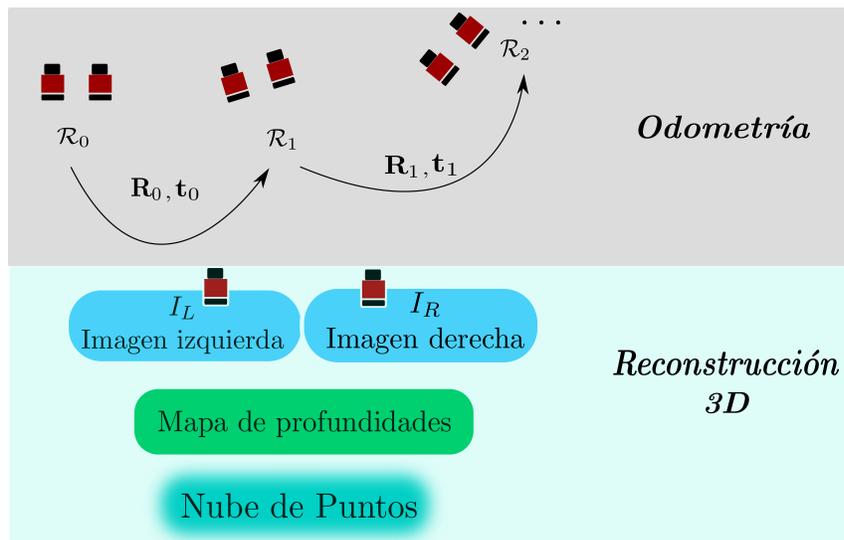


Figura 1.4: Sistema esencial del proyecto.

De forma esquemática los objetivos se muestran en la Figura 1.4. Una vez se logren las, se tiene como otro objetivo aplicar un post-procesamiento a la nube de puntos que genere la reconstrucción 3D. Esto con la finalidad de lograr fluidez en la manipulación del entorno 3D dentro de una interfaz gráfica para la interacción con los resultados.

1.4. Organización del Documento

El manuscrito de esta tesis está dividido en cuatro etapas. La primera etapa es la generación de una secuencia de imágenes en una trayectoria conocida, dentro de ambientes virtuales, haciendo el uso del modelo *pin-hole* [Faugeras, 1993] para cada cámara de un par estéreo. En la misma etapa también se adquiere una secuencia de

imágenes de un par estéreo real, y no se consideran factores como el deslizamiento de las ruedas y la fricción para estimar la trayectoria.

Usando los datos adquiridos en la primera etapa se genera el mapa de disparidades de cada par estéreo en la segunda etapa, y usando los parámetros de calibración se genera la reconstrucción 3D de cada pareja de imágenes. En la tercera etapa se describe el cálculo de la odometría de la cámara basado en homografías [Faugeras, 1993] usando un método de minimización eficiente de segundo orden [Malis, 2004].

A partir de las herramientas descritas, el sistema de este proyecto puede localizarse en un entorno y al mismo tiempo hacer el mapeo 3D del mismo. La visualización de la reconstrucción 3D se describe en una cuarta etapa en donde se desarrolla una interfaz gráfica para observar los resultados e interactuar con ellos. En esta etapa también se realiza un post-procesamiento de los puntos en la reconstrucción 3D con el fin de obtener mejores resultados. En la última parte del manuscrito se presentan las conclusiones y trabajo a futuro.

En el siguiente capítulo se presentan la creación de entornos virtuales para la generación de imágenes en una trayectoria definida y la adquisición de imágenes en entornos reales sobre una trayectoria.

CAPÍTULO 2

Entornos de Navegación

Contenido

2.1. Entorno Virtual	9
---------------------------------------	----------

El entorno es un conjunto de elementos que caracterizan a un sistema físico. Un sistema autónomo, al igual que los humanos, percibe los elementos del entorno en el que se encuentra. Para lograr una navegación eficiente, es necesario conocer los dos tipos de entornos, el real, que es una escena del mundo, y el virtual, que se construye gráficamente por medio de un modelo ideal. El entorno virtual será la referencia en la cual conocemos todos los parámetros del entorno, así es posible medir el desempeño de lo estimado con lo que se conoce idealmente. El entorno real será la base en la cual se tienen factores, como el de un movimiento en el que influyen fuerzas externas o parámetros, como el cambio de iluminación en la escena.

Es necesario mencionar que en los dos tipos de entornos se genera una secuencia de imágenes que será utilizada para el cálculo posterior de la odometría y la reconstrucción 3D.

2.1. Entorno Virtual

CAPÍTULO 3

Estimación de la Odometría y Reconstrucción 3D

Contenido

3.1. Mapa de Profundidad	10
------------------------------------	----

En el Capítulo 2 se habló de la adquisición de las imágenes de un entorno virtual y uno real. En este capítulo se explicará el método para el cálculo de los puntos 3D sobre una trayectoria, usando la geometría para el cálculo del *mapa de profundidades* de cada pareja de imágenes. También se describe el método para obtener la odometría visual con base en la secuencia de las imágenes. En el comienzo de este capítulo se describe la metodología para obtener la profundidad con base en una pareja de imágenes adquiridas desde el par estéreo.

3.1. Mapa de Profundidad

La percepción de la profundidad en la visión humana se basa en la comparación entre las imágenes provenientes de los dos ojos. Fusionando las dos imágenes en el cerebro se obtiene la sensación de la profundidad. Un tema que ilustra esta habilidad de la mente humana son los *estéreogramas*¹.

¹Una imagen bidimensional que cuando es vista apropiadamente crea la impresión de profundidad [Levine and Priester, 2008].

La estéreo visión trata de imitar la percepción de la profundidad. La idea de esto es usar dos cámaras (sistema binocular) que estén dirigidas hacia la misma escena de manera similar a los ojos del humano. La diferencia de la posición entre las cámaras está directamente relacionada con el cálculo de la profundidad. La búsqueda de la relación entre una imagen y otra no es una tarea fácil, pero la complejidad puede ser reducida por análisis geométrico de la configuración del sistema. La geometría para describir la estéreo visión es llamada geometría epipolar.

CAPÍTULO 4

Visualización de la Odometría y Reconstrucción 3D

Contenido

4.1. Estructura de la Interfaz Gráfica	12
--	----

En este Capítulo se describe el desarrollo de una interfaz gráfica que permite visualizar los resultados de la odometría y de la reconstrucción 3D de manera interactiva. También se describe el post-procesamiento de la nube de puntos con la finalidad de obtener mejores resultados del entorno reconstruido. Al final de este capítulo se presentan los resultados finales de este trabajo.

4.1. Estructura de la Interfaz Gráfica

Una interfaz gráfica es una herramienta que permite al usuario interactuar con datos, fórmulas o gráficos. Existen herramientas de desarrollo de interfaces gráficas como Visual Studio[®], GTK, GLADE, entre otras. La mayoría de ellas solo son compatibles con un sistema operativo, en cambio Qt [Dalheimer, 2002; Nokia, 2011], es un sistema de desarrollo de interfaces gráficas en C++ y es la base de este proyecto por su portabilidad entre las tres principales plataformas en la actualidad (Windows, Linux y Mac OS X).

El núcleo de desarrollo de Qt contiene elementos interactivos como botones, etiquetas, sub-ventanas, manejos de cadenas, *widgets*¹ y otros elementos de gran utilidad para lograr la interacción entre humano y computadora (IHC). Una característica de Qt que es atractiva para los desarrolladores es la integración con bibliotecas externas, lo cual también es de gran ayuda para este proyecto debido a que es necesario un motor gráfico para lograr el despliegado de la nube de puntos.

¹Elemento de la interfaz gráfica que despliega información establecida y que puede cambiar el usuario (e.g., caja de edición de texto o una ventana).

CAPÍTULO 5

Conclusiones

En este proyecto de tesis se implementó una metodología para la reconstrucción 3D y la odometría visual. El método fue probado en entornos virtuales y entornos reales. El entorno virtual fue de gran apoyo para comprender cómo es que está compuesto un entorno y para observar las características ideales que deben tomarse en cuenta en el entorno real. Los entornos reales solo fueron adquiridos en un ambiente de interior, es decir con iluminación controlada y sin grandes protuberancias en el suelo. La calibración de las cámaras fue una parte esencial para encontrar los parámetros intrínsecos y extrínsecos del sistema estéreo. Otro elemento fundamental en el entorno real y para la correspondencia de imágenes fue la rectificación de cada pareja de imágenes donde se utiliza el algoritmo basado en [Fusiello et al., 2000].

Una vez que se tenían las adquisiciones de los dos entornos se procedió a realizar la implementación de algoritmos de búsqueda de correspondencia de intensidades. El sistema de búsqueda que se usó fue un sistema robusto y eficiente que, como se menciona en [Geiger et al., 2010], también puede programarse de forma paralela para su implementación en hardware. Este método de correspondencia fue una elección clave para obtener la profundidad de cada pareja de imágenes. Cuando se obtuvo la profundidad se logró encontrar las coordenadas estimadas de los puntos en 3D, a lo cual se le llamó nube de puntos. La profundidad no solo se usó en ese caso, sino que también fue un parámetro primordial para la estimación de la odometría.

La estimación de la odometría se basó en la optimización de una función tal que se obtuvieran los parámetros de rotación y traslación. Los parámetros en los que se basó la función fueron la profundidad y la intensidad de los píxeles de la imagen izquierda. La base para la optimización fue el método de la ESM que resultó ser

un método de minimización que logra que la función de costo converja en pocas iteraciones, lo que significó menor tiempo en la estimación de los parámetros óptimos. Al tener los parámetros que provienen del sistema de estimación de la odometría fue necesario observarlos en conjunción con cada nube de puntos. Esta unión generó la reconstrucción 3D del entorno y para visualizarla se decidió usar una interfaz gráfica que la interacción con la misma.

La interfaz gráfica se diseñó en Qt que resultó ser una herramienta de desarrollo simple de usar e intuitiva. A esta interfaz se le añadió un motor gráfico llamado OGRE, que fue la base en la interfaz para desplegar la reconstrucción 3D y la odometría visual. Una vez que se puede visualizar la reconstrucción 3D se observó que se existían imperfecciones debido al cálculo del mapa de disparidades. Estas imperfecciones se pudieron corregir por medio de una biblioteca llamada PCL que fue de gran ayuda para realizar dos filtrados de las nubes de puntos. La biblioteca de PCL no solo se usó para el filtrado, también fue la base para la triangulación de los puntos para hacer la reconstrucción 3D una malla y no saturar la memoria gráfica desplegando millones de puntos.

El proyecto de tesis se concluyó con buenos resultados como se muestran en los capítulos anteriores, pero el sistema aún no es óptimo. Es posible que se deban considerar entornos reales de exterior y optimizar los procesos para ahorrar tiempo. Esta es la razón por la cual se sugieren trabajos a futuro.

5.1. Trabajo a Futuro

Los trabajos a futuro que se presentan en esta sección se consideran extensiones del mismo, a fin de obtener un sistema más robusto y más completo. Para mejorar el trabajo de este proyecto es necesario considerar los elementos que se listan a continuación:

- *Implementación en GPU.* La implementación en GPU del mapa de disparidades aceleraría el proceso notablemente. Sistemas que se han implementado en GPU logran la aceleración de su sistema hasta cuatro veces con respecto a la implementación en CPU dependiendo de los núcleos utilizados.
- *La autocalibración del sistema estéreo.* Implementar un algoritmo de calibración que permita recalibrar el sistema haciendo uso de la información que adquiere de forma continua. Al implementar la autocalibración se asegura que las mediciones realizadas no serán alteradas por factores como los movimientos bruscos del sistema o cambios de temperatura. Haciendo esto, se obtiene la disparidad y la profundidad con menor error.
- *Considerar cámaras de mayor ángulo.* Las cámaras con un ángulo de visión amplio, como las cámaras panorámicas y las cámaras omnidireccionales, permiten que la exploración dentro de un entorno se realice en menos movimientos que al usar cámaras perspectivas. Con este tipo de cámaras se ahorraría el número de tomas para la reconstrucción 3D de un entorno.

APÉNDICE A

Jacobiano Actual y Jacobiano de Referencia

A.1. Jacobiano Actual

El Jacobiano actual es la derivada de la función de costo que se definió en el Capítulo 3,

$$I'(\mathbf{w}(\hat{\mathbf{T}}\mathbf{T}(\mathbf{x}), Z(u, v), \mathbf{p}(u, v))) - I(\mathbf{p}(u, v))$$

evaluada en $\mathbf{x} = \mathbf{0}$, entonces la expresión para calcular la derivada es la siguiente,

$$\mathbf{J}_c = \nabla_x \{I'(\mathbf{w}(\hat{\mathbf{T}}\mathbf{T}(\mathbf{x}), Z(u, v), \mathbf{p}(u, v))) - I(\mathbf{p}(u, v))\}_{\mathbf{x}=\mathbf{0}}$$

La función I' se puede expandir de la siguiente manera:

$$I'(\mathbf{w}(\hat{\mathbf{T}}\mathbf{T}(\mathbf{x}), Z(u, v), \mathbf{p}(u, v))) = I'(\mathbf{w}(\hat{\mathbf{T}}, Z(u, v), \mathbf{w}^{-1}(\hat{\mathbf{T}}, Z(u, v), \mathbf{w}(\hat{\mathbf{T}}\mathbf{T}(\mathbf{x}), Z(u, v), \mathbf{p}(u, v))))$$

Si se propone $\mathbf{q} = \mathbf{w}^{-1}(\hat{\mathbf{T}}, Z(u, v), \mathbf{w}(\hat{\mathbf{T}}\mathbf{T}(\mathbf{x}), Z(u, v), \mathbf{p}(u, v)))$. Entonces si derivamos I' usando la regla de la cadena obtenemos,

$$\left. \frac{\partial I'(\mathbf{w}(\hat{\mathbf{T}}, Z(u, v), \mathbf{q}))}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}_0} = \left. \frac{\partial I'(\mathbf{w}(\hat{\mathbf{T}}, Z, \mathbf{q}))}{\partial \mathbf{x}} \right|_{\mathbf{q}=\mathbf{p}} \left. \frac{\partial \mathbf{w}^{-1}(\hat{\mathbf{T}}, Z(u, v), \mathbf{w}(\hat{\mathbf{T}}\mathbf{T}(\mathbf{x}), Z(u, v), \mathbf{p}(u, v)))}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{0}}$$

La primer parte de esta derivada es un vector de (1×2) y que puede ser calculado directamente de la información en la imagen actual. En la segunda parte sustituimos $\mathbf{r} = \mathbf{w}(\hat{\mathbf{T}}\mathbf{T}(\mathbf{x}), Z(u, v), \mathbf{p}(u, v))$ y es derivada de la siguiente manera,

$$\left. \frac{\partial \mathbf{w}^{-1}(\hat{\mathbf{T}}, Z(u, v), \mathbf{r})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}_0} = \left. \frac{\partial \mathbf{w}^{-1}(\hat{\mathbf{T}}, Z(u, v), \mathbf{r})}{\partial \mathbf{r}} \right|_{\mathbf{r}=\mathbf{p}'} \left. \frac{\partial \mathbf{w}(\hat{\mathbf{T}}\mathbf{T}(\mathbf{x}), Z(u, v), \mathbf{p}(u, v))}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{0}} \quad (\text{A.1})$$

Ahora, de la primer parte de esta ecuación se puede observar lo siguiente,

$$\left. \frac{\partial \mathbf{w}^{-1}(\hat{\mathbf{T}}, Z(u, v), \mathbf{r})}{\partial \mathbf{r}} \right|_{\mathbf{r}=\mathbf{p}'} = \left(\left. \frac{\partial \mathbf{w}(\hat{\mathbf{T}}, Z(u, v), \mathbf{r})}{\partial \mathbf{r}} \right|_{\mathbf{r}=\mathbf{p}} \right)^{-1}$$

Esta derivada es una matriz de (2×2) la cual representa la variación de coordenadas para un punto \mathbf{p}' en la imagen actual relativa a un punto \mathbf{p} en la imagen de referencia. Para calcular esta derivada es necesario que se tenga bien definida la función *warping*,

$$\mathbf{p}' = \mathbf{w}(\hat{\mathbf{T}}, Z(u, v), \mathbf{p}) = \eta(\gamma(\hat{\mathbf{T}}, Z(u, v), \eta^{-1}(\mathbf{p})))$$

Una vez que se define lo anterior la derivada a ser calculada es,

$$\left. \frac{\partial \mathbf{w}(\hat{\mathbf{T}}, Z(u, v), \mathbf{r})}{\partial \mathbf{r}} \right|_{\mathbf{r}=\mathbf{p}} = \left. \frac{\partial \eta(\gamma(\hat{\mathbf{T}}, Z(u, v), \eta^{-1}(\mathbf{p})))}{\partial \mathbf{r}} \right|_{\mathbf{r}=\mathbf{p}}$$

Entonces, esta derivada se realiza de la siguiente manera,

$$\frac{\partial \eta}{\partial \gamma} \Big|_{\gamma=\mathbf{q}'} \left[\frac{\partial \gamma}{\partial Z} \Big|_{Z=Z} \frac{\partial Z(u, v)}{\partial \mathbf{p}} \Big|_{\mathbf{p}=u, v} + \frac{\partial \gamma}{\partial \eta^{-1}} \Big|_{\eta^{-1}=q} \frac{\partial \eta^{-1}}{\partial \mathbf{p}} \Big|_{\mathbf{p}=u, v} \right]$$

La segunda parte de la Ecuación A.1 al sustituir la definición de la función *warping* es,

$$\frac{\partial \mathbf{w}(\hat{\mathbf{T}}\mathbf{T}(\mathbf{x}), Z(u, v), \mathbf{p}(u, v))}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\mathbf{0}} = \frac{\partial \eta(\gamma(\hat{\mathbf{T}}\mathbf{T}(\mathbf{x}), Z(u, v), \eta^{-1}(\mathbf{p}(u, v))))}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\mathbf{0}}$$

Entonces la derivada de esta segunda parte es calculada usando la regla de la cadena como sigue,

$$\frac{\partial \eta}{\partial \gamma} \Big|_{\gamma=\mathbf{q}'} \frac{\partial \gamma}{\partial \mathbf{T}(\mathbf{x})} \Big|_{\mathbf{T}(\mathbf{x})=\mathbf{I}} \frac{\partial \mathbf{T}(\mathbf{x})}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\mathbf{0}}$$

Al terminar de calcular esta segunda parte de la Ecuación A.1 es posible escribir la expresión para el Jacobiano actual de la siguiente manera,

$$\mathbf{J}_c = \frac{\partial I'}{\partial \mathbf{q}} \left(\frac{\partial \eta}{\partial \gamma} \Big|_{\gamma=\mathbf{q}'} \left[\frac{\partial \gamma}{\partial Z} \Big|_{Z=Z} \frac{\partial Z(u, v)}{\partial \mathbf{p}} \Big|_{\mathbf{p}=u, v} + \frac{\partial \gamma}{\partial \eta^{-1}} \Big|_{\eta^{-1}=q} \frac{\partial \eta^{-1}}{\partial \mathbf{p}} \Big|_{\mathbf{p}=u, v} \right] \right)^{-1} \frac{\partial \eta}{\partial \gamma} \Big|_{\gamma=\mathbf{q}'} \frac{\partial \gamma}{\partial \mathbf{T}(\mathbf{x})} \Big|_{\mathbf{T}(\mathbf{x})=\mathbf{I}} \frac{\partial \mathbf{T}(\mathbf{x})}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\mathbf{0}} \quad (\text{A.2})$$

En la siguiente sección se deduce el Jacobiano de referencia que se calcula con base en una posición de referencia.

A.2. Jacobiano de Referencia

De la misma forma como fue obtenido el Jacobiano actual, el Jacobiano de referencia se calcula por medio de la derivada de la función de costo cuando $\mathbf{x} = \tilde{\mathbf{x}}$. Entonces la expresión para calcular la derivada es la siguiente,

$$\mathbf{J}_r = \nabla_x \{I'(\mathbf{w}(\hat{\mathbf{T}}\mathbf{T}(\mathbf{x}), Z(u, v), \mathbf{p}(u, v))) - I(\mathbf{p}(u, v))\}_{\mathbf{x}=\tilde{\mathbf{x}}}$$

La función I' se define de la siguiente manera:

$$I'(\mathbf{w}(\hat{\mathbf{T}}\mathbf{T}(\mathbf{x}), Z(u, v), \mathbf{p}(u, v))) = I'(\mathbf{w}(\bar{\mathbf{T}}, Z(u, v), \mathbf{q}))$$

Donde $\mathbf{q} = \mathbf{w}^{-1}(\bar{\mathbf{T}}, Z(u, v), \mathbf{w}(\bar{\mathbf{T}}\tilde{\mathbf{T}}^{-1}\mathbf{T}(\mathbf{x}), Z(u, v), \mathbf{p}(u, v)))$. Si el incremento óptimo $\tilde{\mathbf{x}}$ es la solución exacta, entonces es posible verificar que $\tilde{\mathbf{T}} = \hat{\mathbf{T}}^{-1}\bar{\mathbf{T}}$.

Como en el Jacobiano actual, si derivamos I' usando la regla de la cadena obtenemos,

$$\left. \frac{\partial I'(\mathbf{w}(\bar{\mathbf{T}}, Z(u, v), \mathbf{q}))}{\partial \mathbf{x}} \right|_{\mathbf{x}=\tilde{\mathbf{x}}} = \left. \frac{\partial I'(\mathbf{w}(\bar{\mathbf{T}}, Z(u, v), \mathbf{q}))}{\partial \mathbf{x}} \right|_{\mathbf{q}=\mathbf{p}} \left. \frac{\partial \mathbf{w}^{-1}(\bar{\mathbf{T}}, Z(u, v), \mathbf{r})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\tilde{\mathbf{x}}}$$

Una vez más se conoce que la primer parte de esta derivada es un vector de (1×2) y que puede ser calculado directamente de la información en la imagen de referencia. Ahora si $\mathbf{r} = \mathbf{w}(\bar{\mathbf{T}}\tilde{\mathbf{T}}^{-1}\mathbf{T}(\mathbf{x}), Z(u, v), \mathbf{p}(u, v))$ la segunda parte de la derivada una vez más se descompone en dos partes,

$$\left. \frac{\partial \mathbf{w}^{-1}(\bar{\mathbf{T}}, Z(u, v), \mathbf{r})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\tilde{\mathbf{x}}} = \left. \frac{\partial \mathbf{w}^{-1}(\bar{\mathbf{T}}, Z(u, v), \mathbf{r})}{\partial \mathbf{r}} \right|_{\mathbf{r}=\mathbf{p}'} \left. \frac{\partial \mathbf{w}(\bar{\mathbf{T}}\tilde{\mathbf{T}}^{-1}\mathbf{T}(\mathbf{x}), Z(u, v), \mathbf{p}(u, v))}{\partial \mathbf{x}} \right|_{\mathbf{x}=\tilde{\mathbf{x}}} \quad (\text{A.3})$$

Ahora, de la primer parte de esta ecuación se puede observar lo siguiente,

$$\frac{\partial \mathbf{w}^{-1}(\bar{\mathbf{T}}, Z(u, v), \mathbf{r})}{\partial \mathbf{r}} \Big|_{\mathbf{r}=\mathbf{p}'} = \left(\frac{\partial \mathbf{w}(\bar{\mathbf{T}}, Z(u, v), \mathbf{r})}{\partial \mathbf{r}} \Big|_{\mathbf{r}=\mathbf{p}} \right)^{-1}$$

Esta derivada es una matriz de (2×2) , la cual representa la variación de coordenadas para un punto \mathbf{p}' en la imagen actual relativa a un punto \mathbf{p} en la imagen de referencia. Para calcular esta derivada es necesario que se tenga bien definida la función *warping* como en el actual Jacobiano,

$$\mathbf{p}' = \mathbf{w}(\bar{\mathbf{T}}, Z(u, v), \mathbf{p}) = \eta(\gamma(\bar{\mathbf{T}}, Z(u, v), \eta^{-1}(\mathbf{p})))$$

Una vez que se define lo anterior la derivada a ser calculada es,

$$\frac{\partial \mathbf{w}(\bar{\mathbf{T}}, Z(u, v), \mathbf{r})}{\partial \mathbf{r}} \Big|_{\mathbf{r}=\mathbf{p}} = \frac{\partial \eta(\gamma(\bar{\mathbf{T}}, Z(u, v), \eta^{-1}(\mathbf{p})))}{\partial \mathbf{r}} \Big|_{\mathbf{r}=\mathbf{p}}$$

Entonces, esta derivada se realiza de la siguiente manera,

$$\frac{\partial \eta}{\partial \gamma} \Big|_{\gamma=\mathbf{q}'} \left[\frac{\partial \gamma}{\partial Z} \Big|_{Z=Z} \frac{\partial Z(u, v)}{\partial \mathbf{p}} \Big|_{\mathbf{p}=u, v} + \frac{\partial \gamma}{\partial \eta^{-1}} \Big|_{\eta^{-1}=\mathbf{q}} \frac{\partial \eta^{-1}}{\partial \mathbf{p}} \Big|_{\mathbf{p}=u, v} \right]$$

La segunda parte de la Ecuación A.3 al sustituir la definición de la función *warping* es,

$$\frac{\partial \mathbf{w}(\bar{\mathbf{T}}\tilde{\mathbf{T}}^{-1}\mathbf{T}(\mathbf{x}), Z(u, v), \mathbf{p}(u, v))}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\tilde{\mathbf{x}}} = \frac{\partial \eta(\gamma(\bar{\mathbf{T}}\tilde{\mathbf{T}}^{-1}\mathbf{T}(\mathbf{x}), Z(u, v), \eta^{-1}(\mathbf{p}(u, v))))}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\tilde{\mathbf{x}}}$$

Entonces la derivada de esta segunda parte es calculada usando la regla de la cadena como sigue,

$$\left. \frac{\partial \eta}{\partial \gamma} \right|_{\gamma=\mathbf{q}'} \left. \frac{\partial \gamma}{\partial \mathbf{T}(\mathbf{x})} \right|_{\mathbf{T}(\mathbf{x})=\tilde{\mathbf{T}}} \left. \frac{\partial \mathbf{T}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\tilde{\mathbf{x}}}$$

Al terminar de calcular esta segunda parte de la Ecuación A.3 es posible escribir la expresión para el Jacobiano de referencia de la siguiente manera,

$$\mathbf{J}_r = \frac{\partial I}{\partial \mathbf{q}} \left(\left. \frac{\partial \eta}{\partial \gamma} \right|_{\gamma=\mathbf{q}'} \left[\left. \frac{\partial \gamma}{\partial Z} \right|_{Z=Z} \left. \frac{\partial Z(u,v)}{\partial \mathbf{p}} \right|_{\mathbf{p}=u,v} + \left. \frac{\partial \gamma}{\partial \eta^{-1}} \right|_{\eta^{-1}=q} \left. \frac{\partial \eta^{-1}}{\partial \mathbf{p}} \right|_{\mathbf{p}=u,v} \right] \right)^{-1} \left. \frac{\partial \eta}{\partial \gamma} \right|_{\gamma=\mathbf{q}'} \left. \frac{\partial \gamma}{\partial \mathbf{T}(\mathbf{x})} \right|_{\mathbf{T}(\mathbf{x})=\tilde{\mathbf{T}}} \left. \frac{\partial \mathbf{T}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\tilde{\mathbf{x}}} \quad (\text{A.4})$$

Bibliografía

- [Dalheimer, 2002] Dalheimer, M. K. (2002). *Programming with Qt*. O’Reilly, 2nd edition.
- [Faugeras, 1993] Faugeras, O. (1993). *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press.
- [Fusiello et al., 2000] Fusiello, A., Trucco, E., and Verri, A. (2000). A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, pages 16–22.
- [Geiger et al., 2010] Geiger, A., Roser, M., and Urtasun, R. (2010). Efficient Large-Scale Stereo Matching. In *Asian Conference on Computer Vision*, pages 25–38, Queenstown, New Zealand.
- [Geiger et al., 2011] Geiger, A., Ziegler, J., and Stiller, C. (2011). StereoScan: Dense 3d Reconstruction in Real-time. In *IEEE Intelligent Vehicles Symposium*, Baden-Baden, Germany.
- [Giralt et al., 1979] Giralt, G., Sobek, R., and Chatila, R. (1979). Location estimation and uncertainty analysis for mobile robots. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence*, pages 335–338.
- [Lemaire et al., 2007] Lemaire, T., Berger, C., Jung, I.-K., and Lacroix, S. (2007). Vision-based SLAM: Stereo and Monocular Approaches. *International Journal of Computer Vision*, pages 343–364.
- [Levine and Priester, 2008] Levine, G. and Priester, G. W. (2008). *Hidden Treasures: 3-D Stereograms*. Sterling Publishing.
- [Malis, 2004] Malis, E. (2004). Improving vision-based control using efficient second-order minimization techniques. *International Conference on Robotics and Automation*, pages 1843–1848.

- [Matthies, 1989] Matthies, L. (1989). *Dynamic Stereo Vision*. PhD thesis, Carnegie Mellon University.
- [Matthies and Shafer, 1987] Matthies, L. and Shafer, S. A. (1987). Error Modeling in Stereo Navigation. *IEEE Journal of Robotics and Automation*, pages 239–248.
- [Moravec, 1979] Moravec, H. P. (1979). Visual Mapping by a Robot Rover. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence*, pages 598–600.
- [Nistér et al., 2004] Nistér, D., Naroditsky, O., and Bergen, J. (2004). Visual odometry. pages 652–659.
- [Nokia, 2011] Nokia (2011). Qt Developer Network. <http://developer.qt.nokia.com>.
- [Olson and Matthies, 1998] Olson, C. and Matthies, L. (1998). Maximum Likelihood Rover Localization by Matching Range Maps. pages 272–277.
- [Olson et al., 2003] Olson, C., Matthies, L., Schoppers, M., and Maimone, M. W. (2003). Rover navigation using stereo ego-motion. *Robotics and Autonomous Systems*, pages 215–229.