

Capítulo 1

Infraestructura

En este capítulo se detallarán las herramientas base empleadas en la realización de este trabajo.

1.1. Hardware

1.1.1. Sensores RGBD

Los sensores RGBD son sensores capaces de captar a parte de las componentes roja, verde y azul de la luz, información de profundidad (o "D" depth en inglés).

En el 2010 Microsoft publicó el sensor Kinect para la consola de juegos Xbox 360 y Xbox One. Pronto se convirtió en uno de los dispositivos electrónicos más vendidos en todo el mundo después de su lanzamiento.

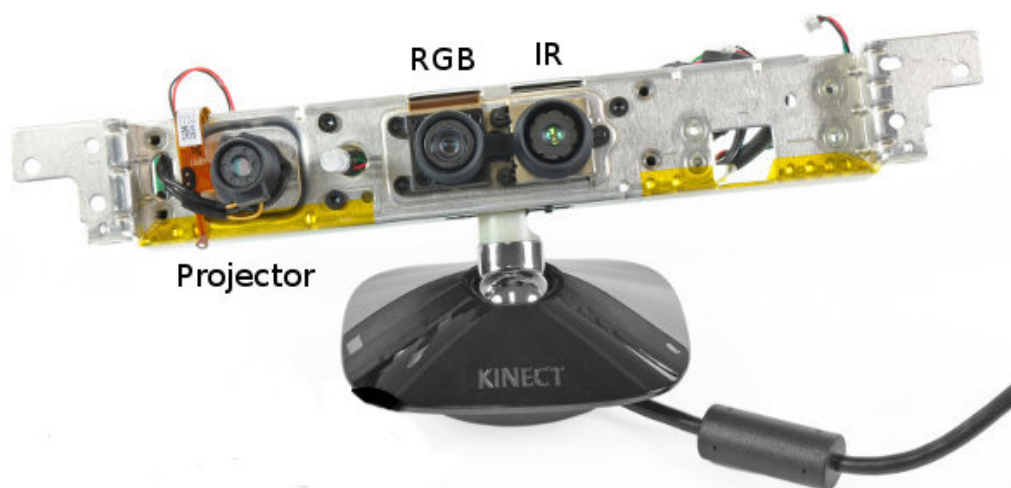


FIGURA 1.1: Sensor Microsoft Kinect

El sensor es capaz de combinar por cada pixel, la información de color con su correspondiente componente de profundidad. Esta tecnología fue desarrollada por la empresa israelí PrimeSense. El sensor Kinect dispone también de un micrófono multiarray con el cual puede predecir de dónde proviene el sonido.

Este sensor salió al mercado a un precio mucho más reducido que algunos que existían antes que él por lo que el interés por este tipo de sensores se disparó y comenzaron a aparecer en diferentes áreas de la tecnología, como interfaces naturales

CUADRO 1.1: Especificaciones técnicas del Asus Xtion PRO LIVE

Campo de visión:	58° H, 45° V, 70° D
Distancia de uso:	Entre 0.8m y 3.5m
Tamaño de la imagen de profundidad:	VGA (640x480) : 30 fps QVGA (320x240): 60 fps
Resolución:	SXGA (1280*1024)

de usuario (en inglés natural user interface, NUI), reconstrucción y realidad virtual o cartografía 3D.

El sensor utilizado en este trabajo es el **Asus Xtion PRO LIVE** que dispone de la misma tecnología comercializado por Asus, que proporciona profundidad, color y audio (utilizando un micrófono multiarray como el sensor Kinect). ¹



FIGURA 1.2: Asus Xtion PRO LIVE

Las especificaciones técnicas de este sensor se encuentran recogidas en la tabla ^{1.1}

1.2. Software

1.2.1. JDeRobot

JDeRobot es un proyecto desarrollado por el grupo de robótica de la Universidad Rey Juan Carlos². Consiste en una plataforma de desarrollo de aplicaciones robóticas y de visión artificial. Está en su mayoría escrito en C++, donde disponen de una colección de componentes capaces de comunicarse a través de ICE middleware³, entre ellos se pueden encontrar herramientas, drivers, interfaces, librerías y tipos.

¹https://www.asus.com/3D-Sensor/Xtion_PRO_LIVE/

²<http://jderobot.org>

³<https://zeroc.com/products/ice>

La versión de JdeRobot empleada ha sido la versión 5.4.0. A continuación se detallarán los componentes de JdeRobot que han sido de utilidad para la realización de este proyecto.

Progeo

Es una biblioteca de geometría proyectiva incluida en JdeRobot, que proporciona funciones muy útiles que relacionan puntos en dos y tres dimensiones.

Será realmente útil en este trabajo para que a partir de puntos en dos dimensiones (píxeles) y su correspondiente información de profundidad (distancia), sacar los puntos relativos de la cámara en tres dimensiones.

Progeo usa el modelo de cámara **Pinhole**, en la Figura 1.3 se puede observar la representación geométrica de la retroproyección y la proyección. Este modelo es definido por unos parámetros intrínsecos y extrínsecos que definen la composición de todos los parámetros iniciales de configuración de la cámara. Los parámetros extrínsecos que establecen la posición 3D, foco de atención (foa) y roll, mientras que los parámetros intrínsecos determinan la distancia focal y el centro óptico o píxel central.

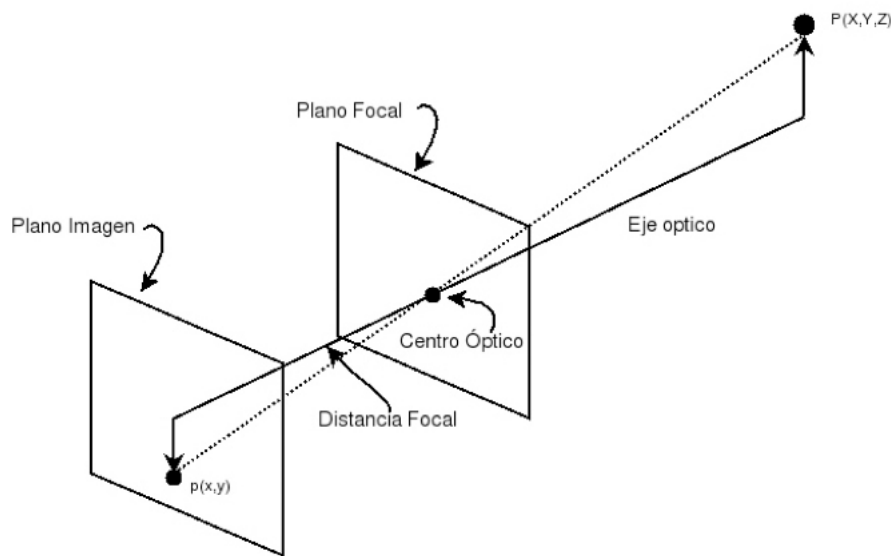


FIGURA 1.3: Modelo de cámara Pinhole

Las funciones que proporciona esta biblioteca son las siguientes:

- **Project:** Esta función permite proyectar un punto 3D del mundo al correspondiente píxel en 2D de la imagen de la cámara.
- **Backproject:** Esta función es capaz de a partir de las coordenadas de un píxel en 2D, obtener la línea de proyección que conecta la cámara y el foco con el rayo 3D que proyecta dicho píxel en el plano imagen. Con esto y conociendo la distancia real del punto 3D a calcular, se calcula las coordenadas reales del punto 3D.

- **DisplayLine:** Esta función permite conocer si una línea definida por dos puntos en 2D es visible dentro del plano imagen.
- **Display_info:** Esta función muestra toda la información sobre la cámara utilizada.

parallelIce

Es otra librería incluida en JdeRobot, que soluciona el problema de latencia de información proveniente de los diferentes drivers, evitando la espera y proveniente de un acceso asíncrono a una copia en local de las interfaces con muy bajo tiempo de procesado.

OpenniServer

OpenniServer es un driver que se comporta como un servidor y es capaz de proporcionar con un sensor RGBD (Kinect o Xtion), imágenes de color, de profundidad o nubes de puntos que son enviados a través de la interfaz ICE a un puerto específico, donde se pueden escuchar los datos. Este driver es el que se necesita para el funcionamiento de este trabajo ya que es desde donde se recogen tanto las imágenes de color (RGB) como las de profundidad (Depth) para su posterior procesado.

RGBDViewer

Es una herramienta que permite enseñar la información proveniente de los sensores RGBD con openniServer como forma de visualización de los datos; imágenes RGB, DEPTH o nubes de puntos.

Pose3D

Es una interfaz que define una posición en tres dimensiones (x, y, z, h) y una orientación con un cuaternión (q0, q1, q2, q3).

RGBPoint

Es una estructura de datos en la cual se pueden guardar puntos en tres dimensiones con coordenadas (x, y, z), color (r, g, b) y un identificador (id). Para generar las nubes de puntos se ha usado un vector de esta estructura.

1.2.2. ICE

ICE (Internet Communications Engine) es un RPC framework desarrollado por ZeroC con soporte en C++, C#, Java, JavaScript y Python entre otros. Se encuentra bajo doble licencia GNU GPL y código cerrado. Actúa como plataforma de comunicaciones y funciona bajo TCP/IP.

En JdeRobot la podemos encontrar como librería y es utilizada como protocolo de comunicaciones entre los diferentes componentes de JdeRobot. En nuestro trabajo

se ha usado para establecer la comunicación entre el componente y el driver del sensor, recogiendo las imágenes de éste.

1.2.3. Point Cloud Library (PCL)

PCL es una librería desarrollada en C++ para el procesamiento de imágenes 2D/3D y nubes de puntos. Está publicada con licencia BSD y libre bajo usos comerciales y de investigación. Está financieramente soportada por un consorcio de compañías comerciales y su propia organización sin ánimo de lucro, Open Perception. A parte de los donadores y contribuidores individuales que aportan al proyecto.⁴

Para simplificar el uso y el desarrollo, esta librería se encuentra dividida en módulos individuales de los que destacan el filtrado de puntos *outliers* o de ruido, estructuras de datos, estimación 3D, algoritmos para la detección de puntos de interés, combinación, segmentación, algoritmos para el reconocimiento de objetos.

En su página web disponen de mucha información y ejemplos prácticos que ayudan mucho a la comprensión de todas las funcionalidades de esta librería. PCL también dispone de una librería i/o de entrada y salida para leer o crear nubes de puntos a partir de diferentes dispositivos, así como visualizadores 3D.

1.2.4. OpenCV

OpenCV (Open Source Computer Vision Library) es una librería de código abierto que fue desarrollada para proporcionar una infraestructura común en aplicaciones de visión artificial y facilitar la inteligencia máquina, con mecanismos de aprendizaje y de interpretación de datos. Con licencia BSD da facilidades para su uso y su modificación bajo fines comerciales.⁵

La librería contiene más de 2500 algoritmos. Estos algoritmos pueden ser usados para detectar y reconocer rostros, identificar objetos, clasificar acciones humanas determinadas en vídeos, seguimiento del movimiento de cámaras, seguimiento de objetos, extraer modelos de objetos 3D, producir nubes de puntos a partir de cámaras, encontrar imágenes similares de un conjunto, juntar trozos de imágenes para producir una imagen final con más resolución, etc... OpenCV tiene más de 47 miles de usuarios en la comunidad, excediendo los 14 millones de descargas, la librería es usada ampliamente en empresas, grupos de investigación y organismos gubernamentales.

OpenCV a sido diseñada de forma eficiente y con un fuerte enfoque en aplicaciones de tiempo real. Escrita en C/C++, la librería obtiene las ventajas del procesamiento multi-núcleo. Dispone de interfaces en C++, C, Python, Java y MATLAB y es soportada por diferentes sistemas operativos como Windows, Linux, Android y Mac OS.

En este trabajo se ha usado a la hora de identificar puntos de interés y para los métodos de emparejamiento. En la Figura 1.4 se puede apreciar un ejemplo de su uso en la detección de puntos de interés sobre un entorno real.

⁴<http://pointclouds.org/>

⁵<http://opencv.org/>

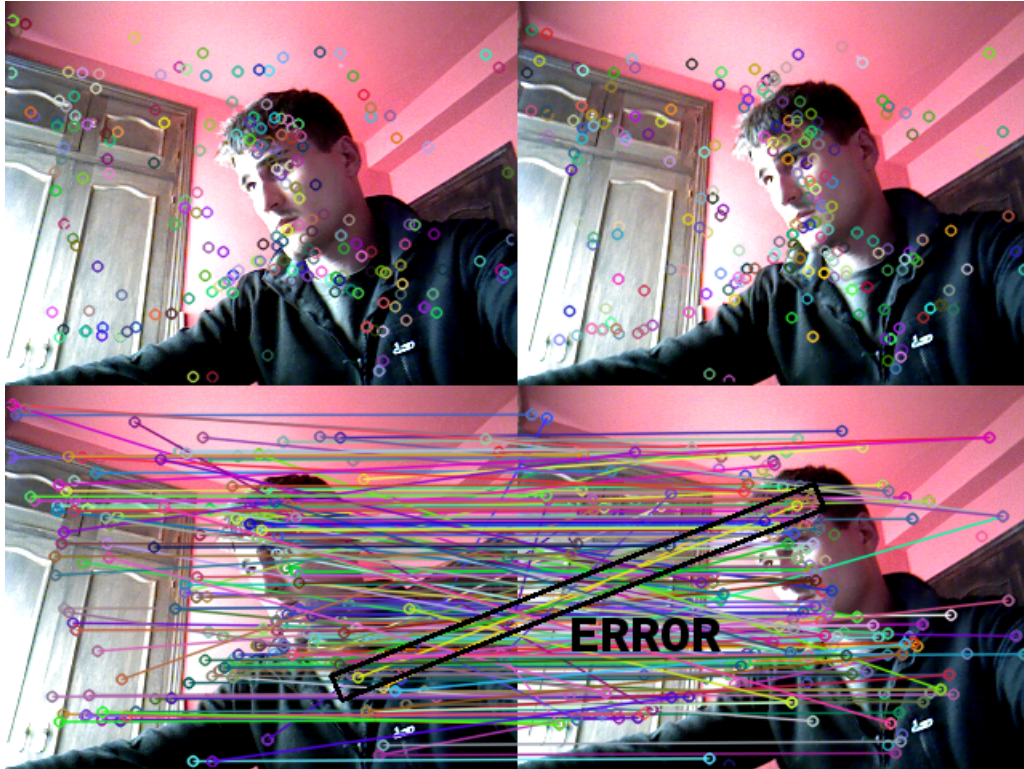


FIGURA 1.4: Detección y emparejamiento de puntos de interés con OpenCV usando SIFT

1.2.5. Eigen

Eigen es una librería de álgebra lineal que permite hacer operaciones aritméticas con matrices y vectores, a través de los operadores comunes de C++, tales como $+$, $-$, $*$ o a través de métodos especiales tales como `dot()`, `cross()`, etc... Para la clase *Matrix* (matrices y vectores) los operadores solo soportan operaciones de álgebra lineal. En Figura 1.5 se puede ver un ejemplo de como de simple es hacer una multiplicación y una división por un escalar.⁶

Eigen es software libre y desde la versión 3.1.1 tiene licencia MPL2 (LGPL3+ para las anteriores versiones). Ha sido usado en este trabajo para realizar los cálculos de la matriz RT a través de los vectores de puntos 3D ya emparejados.

1.2.6. Interfaz gráfica GTK+

GTK+, o the GIMP Toolkit es una herramienta multiplataforma de creación de interfaces gráficas. Es multiplataforma y está escrito en C, pero a sido diseñado para tener soporte para un gran rango de lenguajes, tales como Perl y Python. GTK++ tiene una gran colección de widgets y interfaces para usar en la aplicación, tales como ventanas, botones, selectores, cajas de texto, etc.

⁶<http://eigen.tuxfamily.org/>

Example:	Output:
<pre>#include <iostream> #include <Eigen/Dense> using namespace Eigen; int main() { Matrix2d a; a << 1, 2, 3, 4; Vector3d v(1,2,3); std::cout << "a * 2.5 =\n" << a * 2.5 << std::endl; std::cout << "0.1 * v =\n" << 0.1 * v << std::endl; std::cout << "Doing v *= 2;" << std::endl; v *= 2; std::cout << "Now v =\n" << v << std::endl; }</pre>	<pre>a * 2.5 = 2.5 5 7.5 10 0.1 * v = 0.1 0.2 0.3 Doing v *= 2; Now v = 2 4 6</pre>

FIGURA 1.5: Ejemplo de una multiplicación y división por un escalar con Eigen

Es software libre y parte del proyecto GNU. Con licencia LGPL, permite que sea utilizado por todos los desarrolladores, incluyendo aquellos que están desarrollando software privativo. Ha sido utilizado en muchos proyectos y grandes plataformas.⁷

Glade

Glade es una *RAD tool* (Rapid Application Development Tool) que permite desarrollar de manera fácil y rápida interfaces de usuario en GTK+ para el entorno de escritorio GNOME. La interfaz gráfica diseñada en Glade es guardada en un XML que usando los objetos GTK+ de **GtkBuilder** pueden ser cargados y utilizados por aplicaciones de forma dinámica como se ha hecho en este trabajo.⁸

1.2.7. OpenGL

⁷<https://www.gtk.org/>

⁸<https://glade.gnome.org/>