Capítulo 3

Infraestructura

En este capítulo se explicarán los componentes empleados para desarrollar el trabajo. Tales como el entorno JdeRobot, el simulador Gazebo (con el cual podemos simular las acciones que realizaría un robot en un mundo determinado), la librería de OpenCV (empleada en todo lo relacionado con el tratamiento de imagen), PyQt (para el desarrollo de la interfaz gráfica) y Python como lenguaje de programación.

* 1. Entorno JdeRobot

JdeRobot es una plataforma de software libre para el desarrollo de aplicaciones con robots y visión artificial. Esta plataforma fue desarrollada por el Grupo de Robótica de la Universidad Rey Juan Carlos en 2003 y está licenciada como GPLv3.

JdeRobot está desarrollado en C y C++, aunque contiene componentes desarrollados en lenguajes como Python y Java. El entorno que ofrece está basado en componentes, los cuales se ejecutan como procesos. Dichos componentes operan entre sí a través del middleware de comunicaciones ICE. ICE permite la comunicación entre los componentes incluso estando desarrollados en diferentes lenguajes.

Al tratarse de una interfaz basada en componentes, es capaz de llevar a cabo diferentes tareas en tiempo real de forma sencilla. Cada componente está asociado a un dispositivo hardware del robot, e incluye funciones para poder emplearlo. Esto simplifica el acceso a los diferentes componentes hardware, ya que con una simple función se puede acceder a ellos.

Los robots siempre contienen sensores, con los que se obtienen datos y en función a estos, se envían órdenes a los actuadores. En JdeRobot esta comunicación entre los sensores y los actuadores, es decir, entre los diferentes componentes se realiza mediante ICE. Los datos sensoriales llegan al componente a través de interfaces ICE y los *drivers* encargados de controlar los robots reciben las órdenes también a través de interfaces ICE. En la siguiente imagen se puede ver un ejemplo de esta comunicación con un AR Drone:

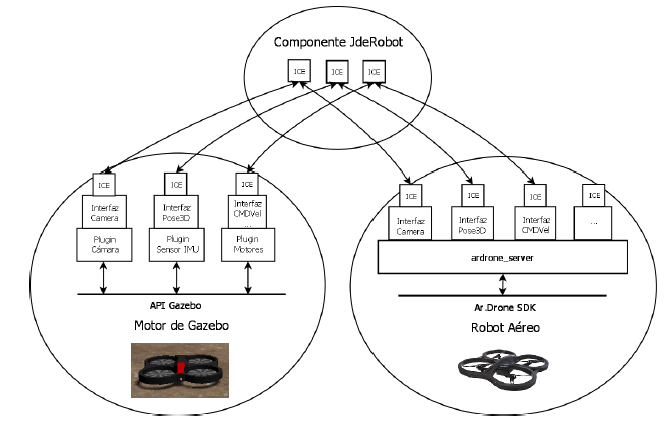


Figure 3.1: Ejemplo de componentes JdeRobot

En el ejemplo de la imagen se puede ver como los componentes de JdeRobot se comunican mediante ICE con los elementos del círculo izquierdo y con los del círculo derecho. En el círculo izquierdo se encuentran los dispositivos del robot simulado en Gazebo, que en este caso son una cámara, un sensor de posición y los motores. En el círculo derecho están los componentes de un robot real. Con esto podemos ver que con JdeRobot se puede teleoperar tanto con un robot simulado como uno real.

Esta plataforma soporta gran variedad de dispositivos como pueden ser el cuadricóptero AR Drone de Parrot, el robot Pioneer de MobileRobotics Inc., el robot Kobuki de Yujin Robot, el humanoide NAO de Aldebaran Robotics, cámaras firewire, USB e IP, los escáneres laser LMS de SICK y URG de Hokuyo, los simuladores Stage y Gazebo, sensores de profundidad como kinect y otros dispositivos X10 de dómotica. A parte de todo esto, tiene soporte para software externo como OpenCV, OpenGL, XForms, GTK, Player y GSL.

En el desarrollo de las prácticas se empleará la versión 5.5.2 de JdeRobot, ya que es la última versión estable.

* 1. Simulador Gazebo

Gazebo es un simulador usado en robótica que permite realizar diversos escenarios tridimensionales donde probar nuestro software. A la hora de realizar el software es necesario hacer pruebas, las cuales saldrían muy caras si se probarán en robots reales (podría no funcionar correctamente y que nuestro robot se rompiera). Por esta razón es muy útil el empleo de simuladores, pues podemos realizar las pruebas que queramos sin peligro de que nuestro robot se estropee. Con los simuladores se pueden diseñar robots y escenarios realistas donde ejecutar nuestros algoritmos.

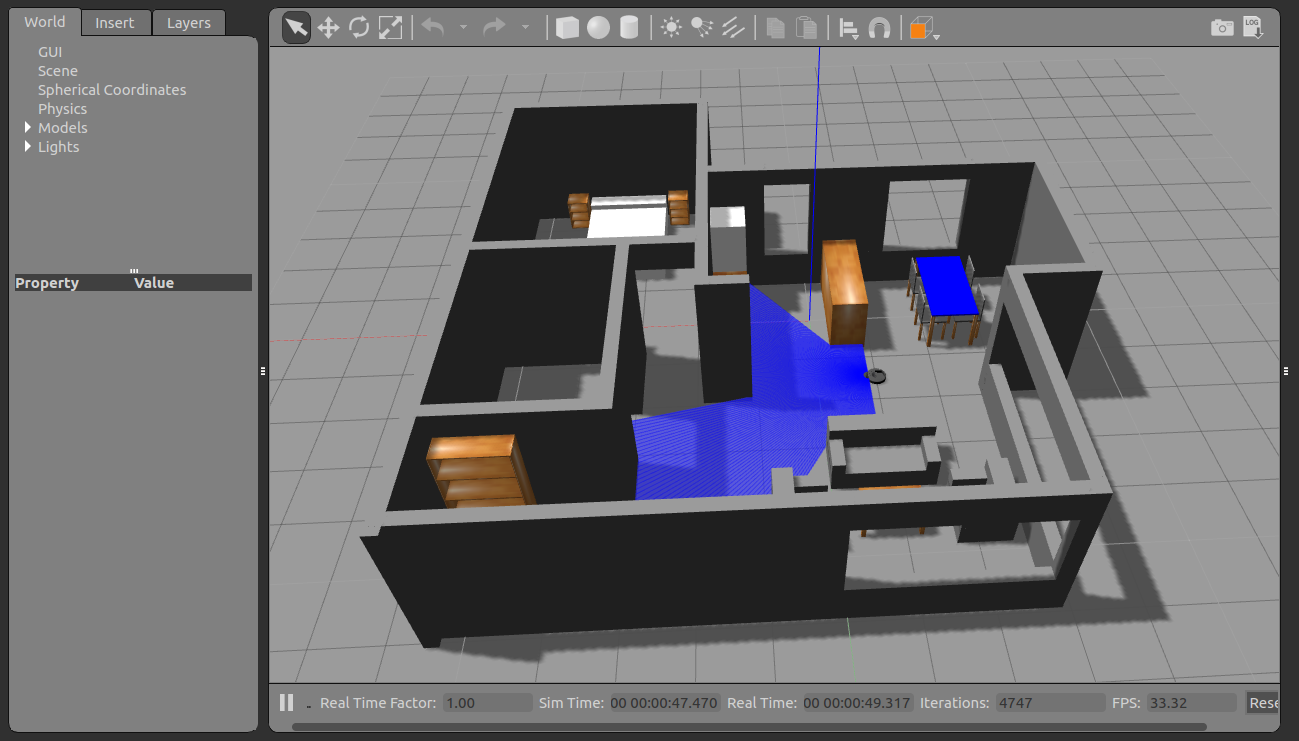


Figure 3.2: Simulador Gazebo

Gazebo es un programa de código abierto distribuido bajo licencia de Apache 2.0. Se emplea en el desarrollo de aplicaciones robóticas y en inteligencia artificial. Es capaz de simular robots, objetos y sensores en entornos complejos de interior y exterior. Tiene gráficos de gran calidad y un robusto motor de física (masa del robot, rozamiento, inercia, amortiguamiento, etc.). Fue elegido para realizar el DARPA Robotics Challenge (2012-2015) y está mantenido por la Fundación Robótica de Código Abierto (OSRF).

En este proyecto se emplea la versión 7 de Gazebo, la cual se usará para crear los diferentes entornos y para probar nuestros algoritmos. Gracias a Gazebo se pueden incluir texturas, luces y sombras en los escenarios, así como simular física como por ejemplo choques, empujes, gravedad, etc. Además, incluye diversos sensores, como pueden ser cámaras y lásers, los cuales podrán ser incorporados en los robots que empleemos. Todo ello hace que sea una herramienta muy potente y de gran ayuda en el mundo de la robótica.

* 1. OpenCV

OpenCV es una librería de código abierto desarrollada por Intel y publicada bajo licencia de BSD. Esta librería implementa gran variedad de herramientas para la interpretación de la imagen. Sus siglas provienen de los términos anglosajones “Open Source Computer Vision Library”, y tal y como se puede deducir, es una librería destinada a aplicaciones de visión por computador en tiempo real.

Esta librería puede ser usada en distintos sistemas operativos como MAC, Windows, Android y Linux, y existen versiones para lenguajes de programación como C#, Python y Java, a pesar de que originalmente era una librería en C/C++. Además, hay interfaces en desarrollo para Ruby, Matlab y otros lenguajes.

OpenCV es una librería que principalmente implementa algoritmos para las técnicas de calibración, detección de rasgos, para el rastreo, análisis de la forma, análisis del movimiento, reconstrucción 3D, segmentación de objetos y reconocimiento. Los algoritmos se basan en estructuras de datos flexibles acopladas con estructuras IPL (Intel Image Processing Library), aprovechándose de la arquitectura de Intel en la optimización de más de la mitad de las funciones.

Incorpora funciones básicas para modelar el fondo, sustraer dicho fondo, generar imágenes de movimiento MHI (Motion History Images), entre otras. Además, incluye funciones para determinar dónde hubo movimiento y en qué dirección.

Fue diseñado para tener una alta eficiencia computacional. Está escrito en C y puede aprovechar las ventajas de los procesadores multinúcleo. La biblioteca de OpenCV contiene más de 500 funciones que abarcan muchas áreas de la visión artificial. También tiene una librería de aprendizaje automático (MLL, Machine Learning Library) destinada al reconocimiento y agrupación de patrones estadísticos.

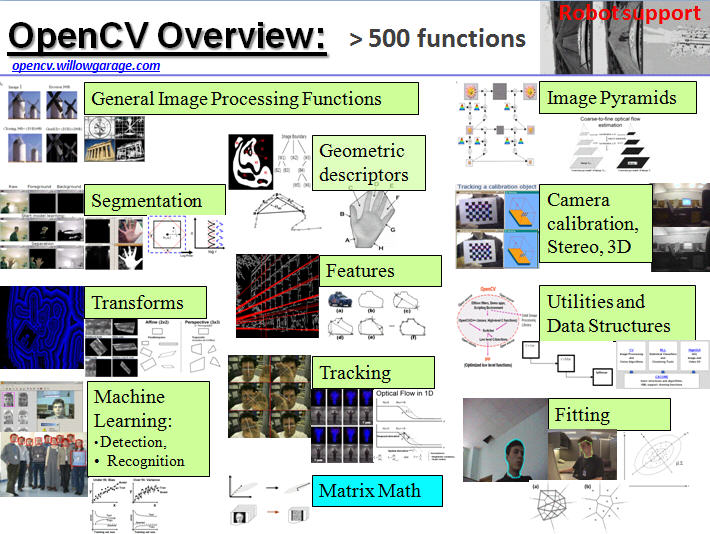


Figura 3.3: Funciones de OpenCV

OpenCV está compuesto por numerosas librerías las cuales nos permiten diferentes acciones como por ejemplo manejar estructuras de datos, detectar bordes y esquinas, escalar o rotar imágenes, modificar el espacio de color de una imagen, realizar matching (comparación de formas para poder saber el parecido entre dos imágenes), detectar líneas y polígonos, tratar objetos en 3D, crear ventanas y asociar eventos a dichas ventanas…

Desde su aparición OpenCV ha sido usado en numerosas aplicaciones entre las cuales se encuentran la unión de imágenes de satélites y de mapas web, la reducción de ruido en imágenes médicas, los sistemas de detección de movimiento, la calibración de cámaras, el manejo de vehículos no tripulados o el reconocimiento de gestos. OpenCV es empleado también en reconocimiento de música y sonido, mediante la aplicación de técnicas de reconocimiento de visión en imágenes de espectrogramas del sonido.

Hay una gran cantidad de empresas y centros de investigación que emplean estas técnicas como IBM, Microsoft, Intel, SONY, Siemens, Google, Stanford, MIT, CMU, Cambridge e INRIA.

En este trabajo se ha empleado la versión 3.2 de OpenCV en Python. Esta librería se empleará para realizar todo lo relacionado con el tratamiento digital de imágenes. Con ello se extraerán datos que puedan emplearse a la hora de tomar decisiones para que los robots funcionen correctamente.

* 1. Python

Python es el lenguaje de programación utilizado en el desarrollo de las distintas prácticas realizadas. Se trata de un lenguaje fácil de aprender y de alto nivel, es decir, se caracteriza por expresar los algoritmos de una manera adecuada a la capacidad [cognitiva](https://es.wikipedia.org/wiki/Cognici%C3%B3n) humana. Su creador fue Guido van Rossum, un investigador holandés que trabajaba en el centro de investigación CWI (Centrum Wiskunde & Informatica). La primera versión surgió en 1991 pero no fue publicado hasta tres años después. Guido dio el nombre de Python en honor a la serie de televisión *Monty Python’s Flying Circus*.

Algunas funcionalidades incluidas en Python son la programación orientada a objetos o el manejo de excepciones, listas y diccionarios. A pesar de todo lo que soporta, se creó con el objetivo de que fuera un lenguaje sencillo de entender, sin perder todas las funcionalidades que pueden ofrecer lenguajes complejos tales como C.

Actualmente Python se trata de un lenguaje de código abierto administrado por Python Software Foundation. Este lenguaje incluye módulos que permiten la entrada y salida de ficheros, sockets, llamadas al sistema e incluso interfaces graficas como GTK y Qt. Además, permite dividir el programa en módulos reutilizables y no es necesario compilarlo, pues es un lenguaje de programación interpretado.

La última versión ofrecida por Python Software Foundation es la 3.6.2, pero en nuestro caso se empleará la 2.7.12.

* 1. PyQt

PyQt es un conjunto de enlaces Python para el conjunto de herramientas Qt, las cuales se emplean para el desarrollo de interfaces gráficas. Fue desarrollado por Riverbank Computing Ltd y es soportado por Windows, Linux, Mac OS/X, iOS y Android.

Qt es un [framework](https://es.wikipedia.org/wiki/Framework) [multiplataforma](https://es.wikipedia.org/wiki/Multiplataforma) [orientado a objetos](https://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos) desarrollado en C++ que permite desarrollar interfaces gráficas e incluye sockets, hilos, Unicode, bases de datos SQL, etc. PyQt combina todas las ventajas de Qt y Python, pues permite emplear todas las funcionalidades ofrecidas por Qt con un lenguaje de programación tan sencillo como Python.

En este proyecto se ha empleado la versión 5 de PyQt. PyQt5 es un conjunto de enlaces Python para Qt5, disponible en Python 2.x y 3.x. Tiene más de 620 clases y 6000 funciones y métodos. PyQt5 dispone de una licencia dual, es decir, los desarrolladores pueden elegir entre una licencia GPL (General Public Licence) o una licencia comercial.

Las clases de PyQt5 se dividen en ciertos módulos, tales como QtCore, QtGui, QtWidgets, QtXml o QtSql. En las prácticas desarrolladas se ha hecho uso de los siguientes módulos:

* QtCore: contiene las funcionalidades principales que no tienen que ver con la interfaz gráfica. Este módulo se emplea para trabajar con archivos, diferentes tipos de datos, hilos, procesos, url, etc.
* QtGui: contiene clases para el desarrollo de ventanas, gráficos 2D, imágenes y texto.
* QtWidgets: dispone de clases que proporcionan un conjunto de elementos de interfaz de usuario para crear interfaces de usuario clásicas de escritorio.