



UNIVERSIDAD DE GUANAJUATO

CAMPUS IRAPUATO-SALAMANCA

DIVISION DE INGENIERIAS

MANUAL DE USUARIO

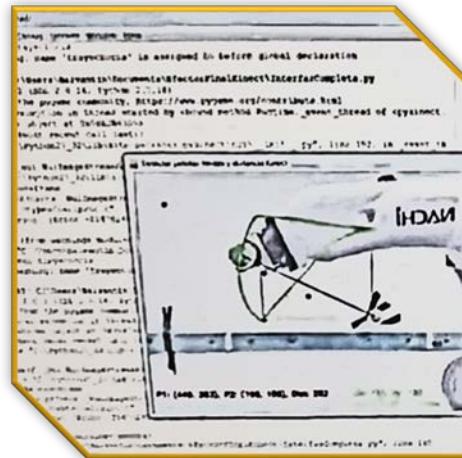
Desarrollo de un sistema de seguimiento de trayectoria para un robot manipulador mediante tecnología accesible de reconocimiento de movimiento

AUTORES:

Cristian Jiménez

Antonio Balvantín

Saúl Hernández



Salamanca, Guanajuato

Esta página se dejó en blanco intencionalmente.

Manual de Usuario

Desarrollo de un sistema de seguimiento de trayectoria para un robot manipulador mediante tecnología accesible de reconocimiento de movimiento

Implementación de Hardware y Software para Tareas de Seguimiento
Versión 1.0 – Julio 2025

1. Información General

Autores: Cristian Jiménez, Antonio Balvantín, Saúl Hernández

Fecha de Publicación: Julio 2025

Versión del Documento: 1.0

Lenguaje de Programación: Python 2.7.18

Tecnologías Clave: OpenCV, Numpy, Kinect, pyKinect

Sistema Operativo Soportado: Windows 7 (todas las ediciones)

Institución: Universidad de Guanajuato – División de Ingenierías Campus Irapuato–Salamanca

Proyecto: Verano de la Ciencia UG 2025 – *Desarrollo de un sistema de seguimiento de trayectoria para un robot manipulador mediante tecnología accesible de reconocimiento de movimiento.*

Correo de Contacto: antonio.balvantin@ugto.mx

2. Objetivo del Sistema

En el presente documento se detalla una guía completa para la instalación, configuración y puesta en marcha de un sistema de seguimiento de trayectorias desarrollado en Python. Este tipo de sistemas se utilizan comúnmente en aplicaciones como detección de objetos, seguimiento de movimiento, clasificación de imágenes y automatización basada en visión computacional.

El objetivo de esta guía es facilitar la implementación del sistema, describiendo paso a paso los requerimientos de hardware y software, así como sus versiones y compatibilidades, para asegurar un entorno funcional y estable. Se abordan aspectos desde la selección del hardware (como cámaras y plataformas de procesamiento), hasta la configuración del entorno Python y librerías específicas como OpenCV, Numpy, entre otras.

Este sistema de seguimiento de trayectoria permite:

- Detectar y ubicar objetos o puntos de interés de un mecanismo en el espacio de trabajo.
- Calcular distancias euclidianas entre puntos de interés, usando marcadores de referencia y seguimiento.
- Realizar un seguimiento de la trayectoria de los puntos de interés en el espacio de trabajo mediante la cámara Kinect y algoritmos de procesamiento de imágenes con OpenCV.

El sistema experimental está diseñado para estudiantes, desarrolladores, investigadores y entusiastas de la inteligencia artificial aplicada, que requieran una solución práctica de visión por computadora con herramientas de código abierto.

3. Requerimientos de Hardware y Software

3.1. Hardware

- **Cámara:** Kinect para Windows con adaptador de alimentación.
- **PC:** Windows 7 (USB 2.0 requerido, USB 3.0 no soportado).
- **Fuente de alimentación:** 127 V AC para el eliminador del Kinect.
- **Espacio libre en disco:** ≥ 2 GB.
- **RAM:** ≥ 4 GB (recomendado 8 GB).

3.2. Software

1. **Windows 7** (cualquier edición).
2. **Kinect for Windows SDK v1.8**
[Descargar aquí.](#)
3. **Kinect for Windows Developer Toolkit v1.8**
[Descargar aquí.](#)
4. **Python 2.7.18**
[Python 2.7.18 Descargar aquí.](#)
5. **Librerías de Python (vía pip):**
 - OpenCV (versión compatible con Python 2.7).
 - Numpy.
 - pyKinect.

(*Buscar versiones compatibles en [PyPI](#).*)



Adaptador



LED de encendido

4. Procedimiento de Instalación

4.1. Instalación de Kinect

1. Conecte el adaptador Kinect a una **fuente AC de 127 V**.
2. Conecte el cable USB del adaptador a un **puerto USB 2.0** en la PC.
3. Conecte la cámara Kinect al adaptador.
4. Verifique que el **LED de encendido del adaptador esté activo** esté activo.
5. Abra el **Administrador de Dispositivos** en Windows y confirme que los controladores estén instalados correctamente.

Si no es reconocido:

- Desconecte y reconecte el cable USB.
- Cambie de puerto USB si es necesario.
- Reinicie el sistema y vuelva a conectar el dispositivo.

Nota: Algunos adaptadores presentan fallas de conectividad; verificar la correcta conexión física e instalación de los drivers.

4.2. Verificación del Kinect

- Ejecute **Kinect for Windows Developer Toolkit v1.8.0**.
- Pruebe alguna de las aplicaciones demo disponibles haciendo clic en **Run** (no es necesario instalar).
 - Se recomiendan los siguientes:

Kinect Fusion Color Basics-WPF

Demonstrates the basics of Kinect Fusion for 3D reconstruction, now including low-resolution color capture.
Please review the documentation for the hardware and software requirements. (C#/WPF sample)

Difficulty: Beginner

Language: C#

 Documentation

 Install

 Run

Skeleton Basics-WPF

Demonstrates the basic scenario of using the SkeletonStream (in default mode or seated mode) and displaying an updated skeleton visualization 30 frames per second. (C#/WPF sample)

Difficulty: Beginner

Language: C#

 Documentation

 Install

 Run

- Si el demo funciona correctamente, el hardware está listo.

4.3. Instalación de Python y Librerías

1. Descargue e instale Python 2.7.18.

```
Name: pykinect
Version: 1.0
Summary: PyKinect Module for interacting with the Kinect SDK
Home-page: http://pytools.codeplex.com/
Author: Microsoft
Author-email: uspython@microsoft.com
License: UNKNOWN
Location: c:\python27_32\lib\site-packages
Requires:
Required-by:
```

2. Abra la consola de comandos (CMD) y ejecute:

1. pip install opencv-python==<versión compatible>
2. pip install numpy==<versión>
3. pip install pyKinect

```
Name: pygame
Version: 2.0.3
Summary: Python Game Development
Home-page: https://www.pygame.org
Author: A community project.
Author-email: pygame@pygame.org
License: GPL
Location: c:\python27_32\lib\site-packages
Requires:
Required-by:
```

3. Verifique la instalación ejecutando:

1. python -m pip list

```
Name: numpy
Version: 1.16.6
Summary: NumPy is the fundamental package for array computing with Python.
Home-page: https://www.numpy.org
Author: Travis E. Oliphant et al.
Author-email: None
License: BSD
Location: c:\python27_32\lib\site-packages
Requires:
Required-by: opencv-python
```

5. Configuración del Sistema

```
#Variables para modificar el programa: -----
#Colores para identificación (rango)
lower_limit = np.array([40, 70, 70])
upper_limit = np.array([80, 255, 255])

#Radio de las esferas:
Radio_size_minimum_detection = 50

#Distancia Kinect al plano de posiciones:
Distance_Kinect_Surface = 71 #Centimetros

#Puntos de la estela:
trail_length = 1800 #Mientras mas grande mas larga y duradera es la estela. Considerese que el sistema funciona a aprox. 60 Hz

#Datos de salida (distancias en centimetros):
#NOTA: Para usar estas variables y exportarlas, basta con usarlas en el ciclo principal y publicarlas a una base de datos u otro programa en python, c++, c o matlab.
Distancia_En_Eje_X = None
Distancia_En_Eje_Y = None
Distancia_Euclidiana = None

#NOTA: No se recomienda alterar otras variables no definidas previamente ya que es muy sensible a cambios y las variables del programa estan basadas en cálculos, direcciones
#de memoria u operaciones necesarias para la compatibilidad del kinect con el sistema.

#-----
```

Abra el archivo principal “*TrackPoint_ProgramaUsuario*” para ajustar los siguientes parámetros:

1. **Colores para identificación (RGB, 8 bits):**
Defina el rango de colores del objeto a detectar.
Ejemplo: para una esfera verde claro, establezca un límite superior brillante y un inferior más oscuro para compensar variaciones de luz.

2. **Radio de las esferas:**

- Un mayor radio discrimina objetos pequeños.
- Recalibre este valor según la distancia Kinect-plano.

3. **Distancia Kinect–plano:**

- Coloque el Kinect **perpendicular** al plano.
- Introduzca la distancia (en cm) en la variable correspondiente.
- Una referencia incorrecta reducirá la precisión.

4. **Estela de puntos:**

- Controla la persistencia de los puntos detectados.
- A mayor número de puntos, mayor duración de la estela.

5. **Variables de salida:**

- Distancia euclídea entre los objetivos (origen y efecto final).
- Distancias en **eje X** y **eje Y**.
- Datos útiles para cálculos de ángulos y trayectorias.

6. Recomendaciones Generales

- Si el sistema presenta **lag** o baja velocidad, conecte la PC a la corriente.
- El sistema fue probado durante **5 horas continuas sin fallas**.
- Identifique el **puerto USB estable** y utilícelo siempre para mayor confiabilidad.
- Evite conectar el Kinect en **USB 3.0**, ya que no es soportado.

7. Historial de Versiones

- **1.0 (Julio 2025)**: Versión inicial de instalación, configuración y puesta en marcha.

8. Contacto y Soporte

- **Autor de correspondencia:** Antonio Balvantín
- **Correo:** antonio.balvantin@ugto.mx
- **Institución:** Universidad de Guanajuato – División de Ingenierías.