# **Applying Visual Odometry for Mapping**

Visual odometry is a technique used to estimate the motion of a vehicle by analyzing the changes in images captured by cameras.

**There are following steps we are going to follow for visual odometry:**

## **1. Hardware Components**

* **Cameras:** Monocular or stereo cameras to capture images of the environment.

**Monocular camera:** A monocular camera, also known as a single-lens camera or single-camera setup, consists of a single lens that captures images of the environment. It operates similarly to the human eye, capturing a 2D image of the scene in front of it.

**Stereo Camera:** A stereo camera setup consists of two or more cameras mounted at a known distance apart, typically capturing images of the same scene simultaneously. By analyzing the differences between the images captured by each camera, stereo vision techniques can be used to estimate depth information.

**Note:** Using Stereo camera is beneficial for SDC as depth is important of images

* **IMU (Inertial Measurement Unit):** To provide additional motion data. accelerometer and gyroscope data to track orientation and movement.
* **Computing Unit:** To process the images and run visual odometry algorithms (e.g., NVIDIA Jetson, Intel NUC).

## **2. Calibration**

* **Intrinsic Calibration:** Determine the camera’s internal parameters (focal length, principal point, distortion coefficients).
* **Extrinsic Calibration:** Determine the position and orientation of the camera relative to the vehicle.

## **3. Visual Odometry Process**

**Image Acquisition**

* + Capture sequential images as the vehicle moves through the environment.

**Feature Detection**

* + Feature detection is a critical step in visual odometry and many computer vision tasks. It involves identifying distinctive key points or features in images that can be reliably tracked over time. Here's a more detailed look at three popular feature detection algorithms: SIFT, SURF, and ORB.

**Feature Matching**

* Match the detected features between consecutive frames to establish correspondences. This can be done using methods like FLANN (Fast Library for Approximate Nearest Neighbors) or brute-force matching.

**Motion Estimation**

* **Essential Matrix (Monocular):** Estimate the relative motion between frames using the essential matrix when camera calibration parameters are known.
* **Triangulation (Stereo):** For stereo cameras, triangulate the matched features to estimate the 3D position of each feature point, then calculate the relative motion.

**Pose Estimation**

* + Calculate the vehicle’s pose (position and orientation) relative to the initial frame using techniques like Perspective-n-Point (PnP) for monocular cameras or by solving the Perspective-3-Point (P3P) problem for stereo cameras.

## **4. Creating the map**

**3D Point Cloud**

* + Accumulate the 3D positions of the tracked features over time to build a point cloud representing the environment.

**Bundle Adjustment**

* + Optimize the positions of the 3D points and the camera poses simultaneously to minimize re-projection error. This step refines the accuracy of the map and the vehicle’s trajectory.

**Loop Closure Detection**

* + Detect when the vehicle revisits a previously mapped area to correct drift and improve the map’s consistency. This can be done using visual place recognition techniques.

## **Algorithms**

**1. SIFT (Scale-Invariant Feature Transform)**

SIFT is designed to detect and describe local features in images. It is invariant to scale, rotation, and partially invariant to changes in illumination and 3D viewpoint.

**2. SURF (Speeded-Up Robust Features)**

SURF is a faster alternative to SIFT. It approximates SIFT's key point detection using Haar wavelet responses and integral images, making it more computationally efficient.

**3. ORB (Oriented FAST and Rotated BRIEF)**

ORB is a fast and efficient alternative to SIFT and SURF, combining the FAST key point detector and BRIEF descriptor with additional improvements for rotation and scale invariance.

## **Comparison**

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature** | **SIFT** | **SURF** | **ORB** |
| **Speed** | Slower | Faster than SIFT | Fastest |
| **Invariance** | Scale, rotation, partial illumination | Scale, rotation, partial illumination | Rotation, partial scale |
| **Descriptor** | 128-dimensional vector | 64-dimensional vector | Binary string (fast matching) |
| **Application** | Accurate, general-purpose | Real-time applications | Real-time, resource-constrained environments |

## **Conclusion**

implementing visual odometry for mapping involves capturing and processing image data to estimate the vehicle's motion and build a map of the environment. By integrating visual odometry , We can create a robust and accurate mapping system for our SDC. Collaboration between groups focused on different aspects, such as odometry, SLAM, and obstacle detection, will ensure a comprehensive solution for real-time navigation and mapping within our college.