Creating a usable map for SDC follows these three general steps:

1. GPS Tour
2. Mapping
3. HD Mapping

**Step 1 — GPS Tour:** The first step involves a preliminary survey of the area using GPS and visual inspection through Google Maps. This step helps identify the main elements and notable challenges in the area.

**Process:**

**1. Planning the Route**:

- Use Google Maps to plan a specific loop or route in the area you wish to map.

- Identify key locations, intersections, and potential challenges like traffic lights, roundabouts, lane merges, pedestrian crossings, etc.

**2. Driving and Data Collection:**

- Equip your vehicle with a GPS device and start driving along the planned route.

- Record GPS data to log the exact coordinates and paths taken.

- Note down the main elements and challenges encountered during the drive:

- Types of roads (residential, highways, etc.).

- Notable landmarks and infrastructure.

- Potential GPS stability issues (e.g., in tunnels or urban canyons).

**3. Notable Elements Mapping:**

- Create a preliminary map by marking the recorded GPS paths and notable elements on Google Maps.

- This map serves as a foundation for the more detailed mapping stages to follow.

**Step 2 — Mapping Format:** Build a functional map that the self-driving car can read and use. This involves selecting the appropriate map format based on the data collected and the intended use of the map.

**Process:**

1. Choose Mapping Type: Based on the requirements, choose the appropriate mapping type (feature, occupancy, point cloud, vector, raster).

2. Data Processing: Use the selected technique and tools to process the GPS and sensor data collected during the GPS Tour.

3. Map Creation: Generate the map in the chosen format, ensuring that it accurately represents the environment and includes all notable elements identified during the GPS Tour.

**Mapping Types and Techniques:**

**1. Feature Maps:**

- Description: Maps created by detecting and mapping features like corners, edges, and points.

- Technique: Use SLAM (Simultaneous Localization and Mapping) algorithms such as ORB-SLAM or RTAB-MAP to convert visual features into a map.

- Tools: Cameras, LIDAR, Visual SLAM software.

**2. Occupancy Maps:**

- Description: Maps that assign values to specific locations indicating whether they are drivable or not.

- Technique: Implement Occupancy Grid Mapping where the environment is divided into a grid, with each cell indicating occupancy.

- Tools: LIDAR, radar, Bird Eye View Networks, Occupancy Networks.

**3. Point Cloud Maps:**

- Description: Maps built using dense collections of points captured by LIDAR.

- Technique: Use SLAM algorithms (e.g., Hector SLAM, GMapping) to process LIDAR data and create detailed 3D point clouds.

- Tools: LIDAR sensors, SLAM software.

**Step 3 — HD Mapping:** Enhance the map by adding high-definition details, including traffic signs, traffic lights, speed limits, road curvature, lane boundaries, and other critical infrastructure.

**Process**:

1. Collect Additional Data: Equip the vehicle with necessary sensors and collect high-resolution data along the mapped route.

2. Annotate Map: Use tools like LGSVL or QGIS to add detailed annotations to the map.

3. Export HD Map: Save the annotated map in a compatible format.

4. Test and Validate: Ensure the HD map is accurate and comprehensive by testing it in the vehicle’s driving system.

**1. Detailed Data Collection:**

- Use additional sensors (high-resolution cameras, more precise LIDAR) to collect detailed data.

- Conduct multiple passes in different conditions to ensure completeness and accuracy.

**2. Annotation:**

- Annotate the map with detailed elements:

- Traffic Signs: Position and type (stop, yield, speed limit, etc.).

- Traffic Lights: Locations and state detection.

- Lane Information: Boundaries, lane markings, and lane types.

- Pedestrian Paths: Crosswalks, sidewalks.

- Curvature: Detailed geometry of roads and turns.

**3. HD Mapping Tools:**

- Use specialized HD mapping software and annotation tools like LGSVL, Autoware, Apollo, or HERE Maps.

- Manually annotate the map or use automated tools to detect and mark features.

- Integrate the HD map with the vehicle’s navigation and perception systems.

**Mapping Techniques:**

1. Visual SLAM
2. Visual Odometry

**Visual SLAM**

Visual SLAM (Simultaneous Localization and Mapping) involves constructing or updating a map of an unknown environment while simultaneously keeping track of an agent's location within it, using visual information (images from cameras). Here’s a breakdown of the workflow into three core parts:

1. Feature Extraction & Visual Odometry

2. Local Mapping & Optimization

3. Loop Closure & Global Optimization

Step 1: Feature Extraction & Visual Odometry

1. Feature Extraction

Feature extraction involves identifying and computing visual features (e.g., corners, edges) from camera images to act as recognizable points. These features are essential for understanding the scene and tracking movement over time.

- Corner Detectors: Identify points where image brightness changes sharply.

- Algorithms: FAST (Features from Accelerated Segment Test), Harris Corner Detector.

- Blob Detectors: Detect regions in the image that differ in properties like brightness or colour.

- Algorithms: SIFT (Scale-Invariant Feature Transform), SURF (Speeded-Up Robust Features).

- Edge Detectors: Find boundaries within images where the brightness changes abruptly.

- Feature Descriptors: Encode feature points into a more compact representation for matching.

- Algorithms: ORB (Oriented FAST and Rotated BRIEF), BRIEF (Binary Robust Independent Elementary Features), BRISK (Binary Robust Invariant Scalable Keypoints).

2. Visual Odometry

Visual odometry estimates the motion of the camera by analyzing the visual features extracted from consecutive frames.

- Feature Matching: Tracks and matches features between consecutive frames.

- Outlier Rejection: Removes incorrectly matched features using methods like RANSAC.

- Optical Flow Estimation: Measures the motion of each pixel from frame to frame.

- Depth Estimation: Estimates the distance of objects from the camera, often using stereo vision.

Step 2: Local Optimization & Mapping

3. Local Mapping: Transforms 2D features into a 3D space, creating a local map of the environment.

- Triangulation: Computes 3D points from 2D feature correspondences across multiple views.

- Stereo Vision: Uses two or more cameras to obtain depth information.

- 3D Reconstruction Techniques: Like Multi-View Stereo and Structure from Motion to build 3D models.

- Prediction and Update: Uses techniques like Kalman Filters or Particle Filters to estimate positions of features and the camera.

4. Local Optimization

Refines the local maps to reduce errors and align them consistently over time.

- Iterative Closest Point (ICP): Aligns point clouds by minimizing the distance between corresponding points.

- Bundle Adjustment: Refines 3D coordinates and camera parameters to minimize the re-projection error.

- Graph Optimization: Constructs and optimizes a graph where nodes represent camera poses and edges represent observations.

Part 3: Loop Closure & Global Optimization

5. Loop Closure Detection

Identifies and corrects revisits to previously mapped areas to prevent drift and accumulate errors over time.

- Bag-Of-Words: Converts images into a set of features that can be matched to previously seen features.

- Graph-Based Approaches: Detects loops by comparing current observations with past observations.

6. Global Optimization

Aligns all local maps into a coherent global map, optimizing the entire trajectory and map.

- Pose Graph Optimization: Represents the entire map as a graph and optimizes the poses.

- Bundle Adjustment: Extends local bundle adjustment to the global scale to refine all camera poses and 3D points.