Summary of MATLAB's Monocular Visual Simultaneous Localization and Mapping (vSLAM)

1. **Map Initialization:** This is the first step and initializes a 3-D map from two frames. Initial correspondences between feature points are identified using the matchFeatures function with the two frames. Then two models, homography and fundamental matrix, are executed over the two frames and the model with the least reprojection error is used. The homography model is ideal for planar scenes while the fundamental matrix is used in general. Once the best model is determined, the relative camera pose is calculated. Using the triangulate function, 3-D coordinates of matched points are computed and a point is valid if it is caught by both frames, has a low reprojection error, and sufficient parallax.
2. **Store Initial Key Frames and Map Points:** The imageviewset object will hold key frames, their attributes, feature points, camera poses, and inter-frame connections while maintaining absolute and relative camera poses. The worldpointset object will hold 3-D map points, their 3-D to 2-D projections, and other attributes.
3. **Initialize Place Recognition Database:** The bag-of-words method is used for loop detection which is an offline visual vocabulary and stored as an object. For our example, MATLAB stored training images to gradually construct a database while simultaneously closing the loop.
4. **Refine and Visualize the Initial Reconstruction:** A brief refinement of the initial reconstruction is performed to optimize camera poses and world points while minimizing reprojection errors.
5. **Tracking:** In the tracking process, each frame is analyzed to decide when to add a new key frame. Features are extracted and matched with known 3-D map points. The camera pose is estimated and refined using various methods. Map points are projected into the current frame for more feature correspondences. Finally, it will decide if the current frame is a new key frame. If tracking is lost due to insufficient matches, the frequency of adding new key frames must be increased.
6. **Local Mapping:** This is executing for every key frame. When a key frame is identified, the attributes of the observed map points are updated. A map point must be observed in at least 3 key frames to minimize outliers. These new map points are created by triangulating feature points in the current and related key frames. Unmatched feature points are matched using the machFeatures function. The local bundle adjustment refines the poses and related key frames.
7. **Loop Closure:** Visually similar images to the current key frame are queried to detect and close loops. A valid loop candidate is found and the relative pose between the candidate and current key frame is computed. The loop connection is added, and mapPointSet and key frames are updated. Finally, a similarity pose graph optimization is performed to correct camera pose drift, and the 3-D map point locations are updated using the optimized poses and scales.
8. **Compare with the Ground Truth:** Finally, they compare the optimized camera trajectory with the ground truth, which is given in a text file. For higher resolutions, the numPoints variable must be increased which will also increase computation time. It is also important to be aware of the frame rate since tracking can be lost if the camera motion is too fast, losing connections between key frames.

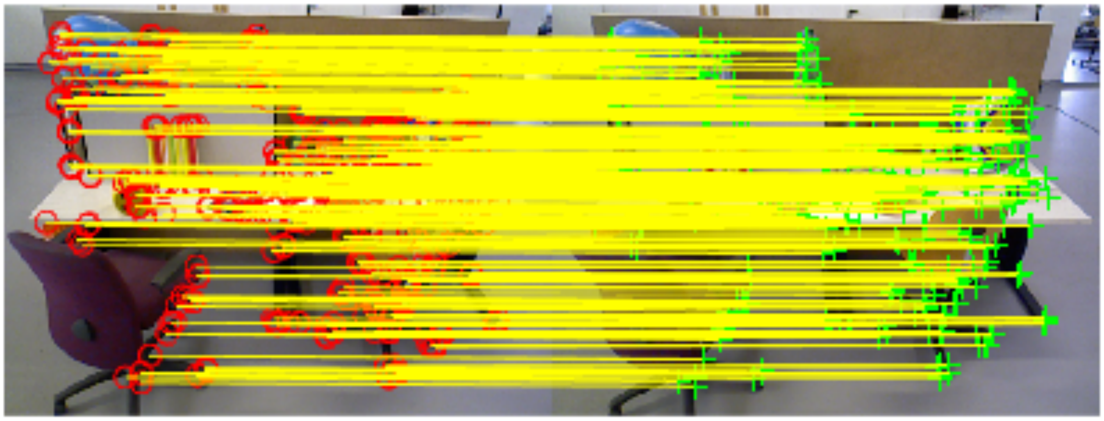

Figure 1: Map Initialization

Figure 2:Visualize Initial Reconstruction

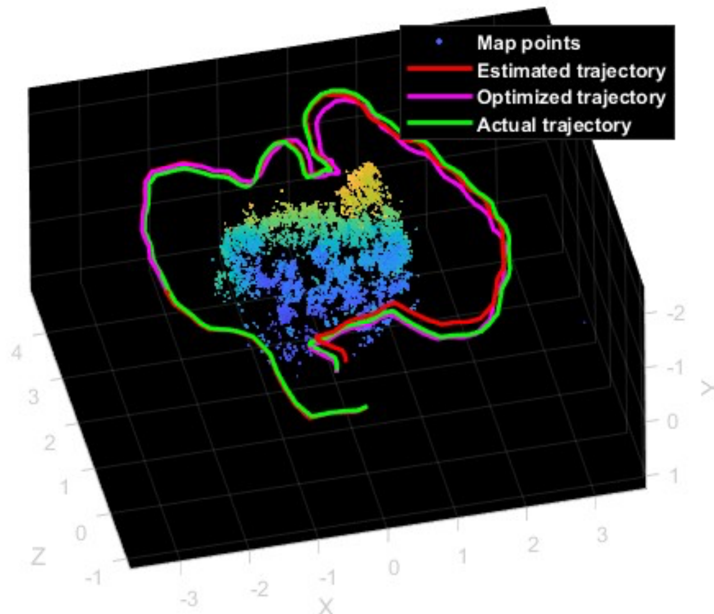

Figure 3: Final Point Cloud and Trajectories