Online object detection and localization on stereo visual slam system

Saturday, December 26, 2020 4:17 PM

The system processes a stream of stereo images and builds an object map with semantic annotation online.

Each object in the map is represented by its oriented bounding box (OBB), that has
- position T
- orientation R
- dimensions D

The method can be split into three modules
- the S-PTAM module
- RCNN module to detect objects in the left image.
- Object mapping module to estimate and refine the location of the objects.

RCNN module for object detection.
- Faster RCNN is used. Extra layers are added to predict the object rotation R and dimension D. (original RCNN provides class and bounding box).
- For orientation R, set pitch and roll to zero, only consider yaw.
  - The CNN predicts the yaw angle between the camera principle ray, and the ray that cross the bounding box center.
  - The orientation is discretized into n bins, and the net predicts a confidence for each bin, indicating the probability that the angle lies inside it, and a residual correction angle to the center of the bin, given by its sine and cosine.

The list of detections by the CNN are sent to the object mapping module, where the object poses are estimated relative to nearest keyframe processed by S-PTAM.
- This involves calculating the OBB from the 2D bounding box, orientation, and size,
- Estimated object position is further refined by using the data from the S-PTAM point cloud.
- After performing data association with the objects already present in the map, data fusion is performed to update the map.

Rough object pose estimation
Combining estimates of the object R, D and 2D bounding box enables us to predict the object's 3D bounding box, as presented in "3D bounding box estimation using deep learning and geometry". The method is based on the fact that the projection of the bounding cube should fit tightly within the bounding box.

Once the initial pose estimation of the object is computed, we use the estimated pose by S-PTAM to compute the object poses in the map frame.

Object matching.
The object correspondence is achieved by considering the IoU between the bounding box of a new detection B_d and each map object projected into the image plane b_p. By choosing the map object which maximizes the IoU, we match the objets. If the IoU is not greater than a threshold for every object, we add a new object.

Object pose refinement.

When the 3D pose of a detected object is estimated by means of its orientation, dimensions and 2D bounding box, it can have a considerable error. That is, the predicted object's distance to the camera is sensitive to small errors in the estimated bounding box and object dimensions. In

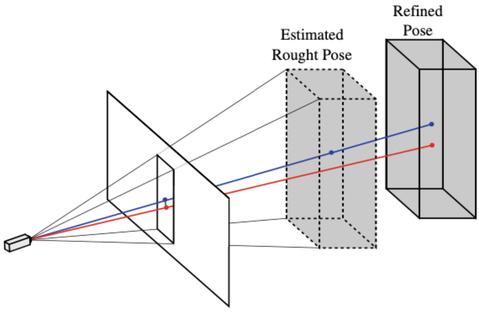

Fig. 4 Objects pose refinement using the feature - map point matches obtained by S-PTAM. The blue dot in the image plane is the projection of the centroid of the Rough Estimated Object (blue dot in the 3D space closest to the camera). The red dot in the image plane is the feature closest (in terms of euclidean distance) to the projection of the centroid. The depth of the 3D map point (red dot in the space) associated to the feature is used to set the refined pose of the object centroid (blue dot in the 3D space farthest to the camera)

Fusion of object observations.

If $\boldsymbol{O} = \{O^t\}$ is the set of objects, for each object O^t we can consider its observations $\boldsymbol{q}^t = \{q_i^t\}$. Every observation q_i^t is parameterized by a 3D position $T_i^t = (x_i^t, y_i^t, z_i^t)$, an orientation given by the angle θ_i^t, the OBB dimensions $D_i^t = (dx_i^t, dy_i^t, dz_i^t)$ and the object category c_i^t. The pose is represented relative to the nearest keyframe of the observation. Fusion of all these observations is performed by obtaining the median of each parameter, at a given moment, the object's O^t position $T^t = (X^t, Y^t, Z^t)$ is determined by:

$$\begin{bmatrix} X^t \\ Y^t \\ Z^t \end{bmatrix} = \begin{bmatrix} \text{median}\,(x_i^t) \\ \text{median}\,(y_i^t) \\ \text{median}\,(z_i^t) \end{bmatrix}. \quad (2)$$

The object dimensions $D^t = \text{median}\,(D_i^t)$ are computed similarly. Observe that these fusion strategy are valid for an environment where all objects are stationary.