FroDO From detection to 3D objects

Saturday, December 26, 2020 10:40 AM

Purpose:
3D object detection:
input: point clouds
output: 3D bounding boxes enclosing the objects.

Approach:
- Detection
- Selection
- Shape encoding
- Pose and shape optimization.

input: RGB images.
Detection: use an off-the-shelf detector to detect objects.
Selection: connect the same object across different frames.
Shape encoding: use CNN to extract a 64D vector from the images that belong to the same object.
Optimization: optimize the shape and pose.

Detection:
- Use mask rcnn to obtain object bounding box and mask M.

Selection:
- Similar to ImVoteNet.
- The ray from the center of the 2D bounding boxes also coincide at the center of the 3D object.
- Use DP-means (similar to k-means) to find the 2D bounding boxes whose rays coincide, and use the reprojection of the 3D bounding box to remove the 2D bounding boxes that have small IOU.

Shape encoding:
- feed the 2D image of an object to the shape encoding module.
- output a 64D vector from each image to represent object shape.
- baseline encoder is resnet,
- merge all the 64D vectors for the same object.
  - first method, take average.
  - Second method, majority voting to find the nearest neighbor in the dataset.

Optimization.
- for each object k, use the 64D vector as input, feed to a shape decoder to obtain a sparse point cloud,
- the pose is from the 3D bounding box from selection part,
- brute-force search for the rotation matrix for the object to be on the ground.
- use latent vector and point position, feed to DeepSDF to obtain the dense representation,
- define training loss (energy):
  - 2D silhouette loss E_s,
  - photometric consistency loss E_p.
  - geometry loss E_g
  - shape regularization loss E_r.

The losses are divided into sparse and dense losses.
Sparse loss.
- 2D silhouette loss E_s.

Defined as the difference between the 3D shape reprojection and the 2D mask from the mask rcnn.

$$E_s(z_k, T_{wo}^k) = D_c(M, \pi(T_{cw} T_{wo}^k G(z)))$$

- M is the mask sample.
- T_{cw} is camera pose, from world to camera frame.
- T_{wo} is the object pose from the 3D bounding box, from object to world frame.
- $\pi(x)$ is the reprojection of 3D point cloud to 2D.
- $G(\cdot)$ is the function that reconstructs the 3D sparse point cloud using the 64D vector, i.e. a decoder.
- D_c is a distance measure.

- photometric loss E_p, makes the color of the one 3D point stays the same in the multi-view images,

$$E_p(X, I^R, I_1^S, \dots I_N^S) = \frac{1}{N \cdot |X|} \sum_{i=1} \sum_{x \in X} \| r(I^R, I_i^S) \|_h$$

$$r(I^R, I^S) = I^R(\pi(T_{cw}^R x)) - I^S(\pi(T_{cw}^S x))$$

- X is the point cloud.
- Project points in X to N neighboring frames, compare the photometric difference.
- I^S is obtained from reprojecting X,
- I^R is the reference frame,

- geometry loss E_g,

minimize the predicted point cloud and GT. point cloud from SLAM.

$$E_g(z_k, T_{wo}^k) = D_c(X_{slam}, T_{wo}^k G(z)),$$

Dense loss.
- E_p, E_g same with sparse loss.
- E_s: use the foreground/background probability for 2D points.

$$E_s = \int_\Omega H(\phi) P_f(x) + (1 - H(\phi)) P_b(x) \, d\Omega.$$

$$H = 1 - \exp \prod_{x \text{ on ray}} (1 - \mathrm{sig}(\zeta \cdot \phi(x))),$$

- ϕ is 3D/2D shape.
- H is a mapping to the 2D image, i.e. object mask.
- ζ is a smoothing function, $1 - \mathrm{sig}(\zeta \cdot \phi(x))$ is the probability to get point x in the background
- P_b is the observed probability of point x in 2D. to be the background.
- P_f is the probability that 2D point x is in the foreground.

Summary.
- spare, dense energy is quite novel.
- 64D vector merge, and the clustering using DP-means can be improved.