MO-LTR

Thursday, December 17, 2020 10:36 AM

Intro:
This paper is concerned with the problem of online detection, tracking and reconstruction of potentially dynamic objects from monocular videos.
- online } FroDo is offline, batch processing, and assumes static objects.
- dynamic
- monocular ← do not use depth, but use shape prior instead.

Review:
Learning a shape prior that takes advantage of object shape regularity is another research trend for object shape reconstruction.
Intra-class full 3D shape variance is captured in a learned latent space, Object shape is optimized in this latent space given image or depth evidence, and then full 3D objects can be reconstructed even if only partial observation is available.

Leveraging a ray clustering based approach for data association, FroDo demonstrates multi-object reconstruction from monocular image sequences.
- Although FroDo and the proposed system share common ground on following coarse-to-fine reconstruction, where objects are firstly localized and represented coarsely using cubes/ellipsoids, followed by a dense shape reconstruction, the ray clustering algorithm of FroDo assumes a static environment, and is offline.

Method
- Given a new RGB frame, MO-LTR first employ a monocular 3D detector to predict a 9-DoF object pose, object class label, and 2D bounding box.
- For each detected object, an image patch cropped by the 2D bounding box of an object is mapped to a shape code in a latent shape embedding.
- State (pose and motion) of each existing object in the map is modelled by a multiple Bayesian filter.
- Prior to data association, we use the filter to predict object location and decide whether an object is matchable using the predicted motion status.
- The new detections are associated to the matchable objects based on simple but practical pairwise cost (i.e. 3D Generalized IOU) as the matching cost.
- We solve the linear assignment problem using the Munkres algorithm to decide whether a detection merges to an object track or instantiate a new object in the map.
- Filters are updated using the associated detections.
- To reconstruct an object shape, multiple single-view shape codes are fused into a single one by taking the mean, which is decoded by the shape decoder to a TSDF.
- The object shape is transformed to the world coordinate using the updated object pose.

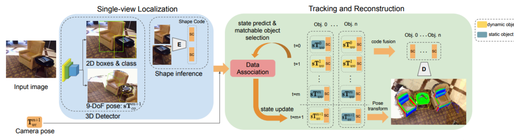

Fig. 2: MO-LTR pipeline. Given a new RGB frame, we predict 6-DoF object pose with respect to the camera $\mathbf{T}_{co}^{m+1}$ and object scale $\mathbf{s}$ (*i.e.* 3D dimension) for each object of interest, which is visualized as an oriented 3D bounding box. we also predict object class and 2D bounding box for each object. An image patch cropped by the 2D bounding box is mapped to a single-view shape code via the shape encoder. The state of objects in the map is tracked by a multiple model Bayesian filter. The motion status is indicated by different background colors of object poses. After filter prediction, matchable objects are used to associate to the new set of detections. A matched detection is attached to the object track, and the shape is progressively reconstructed by decoding the fused shape codes.

3D localization.
MO-LTR detects objects of interest given a RGB image.
- A monocular 3D detector that takes a single RGB image as input, and outputs both 2D attributes (i.e. object class and 2D object bounding box), and 3D attributes (i.e. object translation t_{co} and viewpoint R_{co} wrt camera, and object 3D dimensions (S_x, S_y, S_z)).
- The detector is trained to predict an offset $(\delta x, \delta y)$ between the center of the 2D bounding box (x_{2d}, y_{2d}), and the projection center of the 3D shape (x_{3d}, y_{3d}), on the image.
- We also predict the object depth z.
- Assuming we know the camera intrinsic parameters f_x, f_y, c_x, c_y, the object's 3D center t_{co} in camera frame is:

$$t_{co} = \left(\frac{x_{2d} + \delta x - c_x}{f_x} z, \frac{y_{2d} + \delta y - c_y}{f_y} z, z \right).$$

- To predict object viewpoint, azimuth R_{azi} and elevation R_{ele} are discretized into 36 and 10 bins respectively, the rotation matrix is

$$R_{co} = R_{ele} R_{azi}.$$

- $T_{co} \in SE(3)$ from the canonical object space to camera frame is:

$$T_{co} = \begin{bmatrix} R_{co} & t_{co} \\ 0 & 1 \end{bmatrix}$$

- Together with the scale parameters, each detected object is localized as an oriented 3D bounding box in camera frame.

Shape embedding and inference.
The formulation follows FroDO.
- We use a compact K-dimension shape code $L \in \mathbb{R}^K$, embedded in a learnt latent space to parameterize normalized object shapes in a canonical pose.
- This learnt latent space is a shape prior.
- A TSDF, where the zero-crossing level set is the object surface, can be decoded from the latent code via a DeepSDF decoder $G(L)$.
- After each object detection, we estimate a single-view shape code by mapping its cropped 2D bounding box to the shape embedding using the shape encoder.
- Shape is decoded later once shape codes are fused over time.

Tracking
- Interacting multiple model filter
- Munkres algorithm

Reconstruction.
After object tracking, the last step is to reconstruct a dense shape of each object in map.
- We fuse all single view shape codes up to the current frame by averaging them into a single code

$$L_f = \frac{1}{N} \sum_i^N L_i$$

- A TSDF representing object shape in the canonical object coordinate is decoded from the shape decoder

$$X^o = G(L_f)$$

- An object shape mesh is extracted from the TSDF by the marching cube algorithm.
- The mesh is transformed to world frame using

$$X^w = T_{wo} S X^o, \quad S = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & S_z \end{bmatrix}$$

T_{wo} is the transformation from object frame to world frame.
S is the scale matrix.

Implementation ← same with FroDO
- We use CAD model annotation provided by Scan2CAD.
- Outdoor 3D object detection uses "Tracking Objects as Points", → CenterTrack: a monocular 3D multiple object tracking framework.
- We use K=64 dimensions for shape embedding. The architecture of the shape decoder is same with DeepSDF.
- We use ground-truth camera pose in KITTI and ScanNet.

Experiments
- We compare MO-LTR with FroDo on object localization rather than shape reconstruction, using mAP, with IoU threshold = 0.5.
- FroDo uses 2D detections and a ray clustering approach for data association. The 3D bounding boxes are obtained by triangulating associated 2D detections. The ray clustering based data association suffers from local minima and leads to incorrect matching if objects are close to each other.
- MO-LTR circumvent this problem by using monocular 3D detection and then the following data association works in the 3D space directly.
- We compare MO-LTR wrt MOTSFusion, where they use a monocular depth estimation framework followed by an instance segmentation network for object reconstruction to recover the visible surface of objects.