Fusion aware conv

Sunday, January 10, 2021 8:19 AM

The input is an online acquired RGB-D sequence being reconstructed with real-time depth-fusion, denote RGBD sequence by

$$f^k = \{(C_m^k, P_m^k)\}_{m=0}^{M}, k = 0 \cdots K$$

where C_m^k, P_m^k store RGBD info of pixel m of frame K and its 3D coordinates.

Given a reconstruction represented by a point set P, we construct a global tree T_G maintaining the spatial organization of all points, as well as a per-point local tree $\{T_L(p)\}_{p \in P}$ storing the 1-ring neighbourhood for each point.

Global coordinate interval tree.
We maintain three coordinate interval trees T_G^x, T_G^y, T_G^z, one for each dimension.
Eg. For T_G^x, each node $n_i \in T_G^x$ records a set of points $P_i^x \subset P$ in which each point has its X-coordinate lie in interval $[X_{min}(n_i), X_{max}(n_i)]$. We stipulate the adjacent nodes in a coordinate interval tree complies with constraints:

$$X_{max}(n_l) < X_{min}(n_p),\ X_{max}(n_p) < X_{min}(n_r).$$

with n_l, n_r being the left and right child of n_p. The entire 3D scene is then split into slices along x-dimension.

Point insertion of coordinate interval tree
Given a 3D point $P = (x_p, y_p, z_p)$, we first find a node $n_i \in T_G^x$ satisfying

$$x_p \in [X_{min}(n_i), X_{max}(n_i)]$$

through a top down traverse of tree. If such a node exists, p is added to point set P_i^x of the node. Otherwise, we create a new leaf node whose point set is initialized as $\{p\}$ and coordinate interval as $[x_p - h, x_p + h]$.

After constructing the coordinate interval trees for all three dimensions, we can achieve efficient point correspondence search and neighborhood retrival for any given query 3D point $q = (x_q, y_q, z_q)$. Through traversing the three trees, we obtain three nodes

$$n_i \in T_G^x,\ n_j \in T_G^y,\ n_k \in T_G^z$$

satisfying

$$x_p \in [X_{min}(n_i), X_{max}(n_i)]$$
$$y_p \in [y_{min}(n_j), y_{max}(n_j)]$$
$$z_p \in [z_{min}(n_k), z_{max}(n_k)]$$

The neighbouring points are simply the intersection of the three corresponding point sets:

$$\mathcal{N}(p) = P_i^x \cap P_j^y \cap P_k^z.$$

Local per-point octrees
We need to sort the set of neighboring points into a structured organization based on surface aware metric. This is achieved by maintaining per-point octrees so that the surface-aware neighborhood in arbitrary scale can be found efficiently.

Given point $p \in P$, first retreive its local neighborhood $\mathcal{N}(p)$ using coordinate interval trees. Then divide the extended point set $\cup \mathcal{N}(p)$ according to eight quadrants of Cartesian coordinate originated at P.

Within each quadrant, we add the point that is closest to p as the child of the corresponding direction, we then compute a 1-ring neighbourhood in a direction-aware octree. One can easily expend the 1-ring neighborhood of a point into multiple rings through chaining octree-based neighbor searches.

Fusion aware point convolution.
We propose a convolution operation which extends PointConv with intra-frame and inter-frame feature fusion named fusion aware point convolution.

$$PC_p(W, F) = \sum_{\Delta p \in \Omega} W(\Delta p) F(p + \Delta p)$$

where $F(p + \Delta p)$ is the feature of a point in local region Ω, centered at p, W is weight.

2D-3D feature fusion.
The feature coding at a 3D point should consider all matched pixels in different frames. Pixel correspondence between consecutive frames can be easily retrieved based on T_G^x, T_G^y, T_G^z. Each 3D point p has a set of 2D pixels

$$I(p) = \{c^k \mid k \in n\}$$

We can extract feature for each pixel c^k intra-frame via 2D conv. We adopt FuseNet as 2D feature encoder

$$F^{2D}(c^k) = FuseNet(f_k, c^k)$$

where $FuseNet(f_k, c^k)$ is 2D feature for pixel c^k in frame f_k.
∴ Each 3D point p in the scene has a set of corresponding 2D features, and max pooling is adopted to fuse them into one feature.

$$F^{2D3D}(p) = maxpooling\{F^{2D}(c^k) \mid c^k \in I(p)\}.$$

Octree induced surface aware 3D convolution.
The local region Ω for each point p is given by octree $T_L(p)$, this could ensure the neighborhood would only enlarge along the object surface, which is surface-aware. but not skip some gaps to reach shortest distance.

Frame to frame feature fusion.
We use the segmentation results given by previous frames to improve the performance of following frames. For each 3D point P, our method would update its segmentation result if it is observed by a new frame f_i, if p has low segmentation uncertainty in frame f_i, the current form of feature fusion should be useful in future prediction, the fusion is conducted via max-pooling.