OctNet: Learning Deep 3D Representations at High Resolutions

Motivation
In 3D convolutional networks memory requirements increase cubically w.r.t. input resolution.

3D data is usually very sparse.

Data structure: grid of shallow octrees [1]
- Shallow octrees have fixed depth
- Efficiently encoded with bit-string
- Bit indicates if node is split or not
- Fast neighbour address resolution via bit operations

Motivation
Combines shallow octrees
Data on fixed resolution is pooled
Convolution pools data in larger octree cells

Combines shallow octrees
Data on fixed resolution is pooled

Combines shallow octrees
Data on fixed resolution is pooled

Combines shallow octrees
Data on fixed resolution is pooled

Convolution pools data in larger octree cells

Accurate OctNet Convolution

$O(i, j, k) = \sum_{j=0}^{L_1} \sum_{k=0}^{L_2} \sum_{p=0}^{L_3} T_i(j,m,n) \cdot O(p, m, n)$

Efficient implementation

Fast neighbour address resolution via bit operations
- Bit indicates if node is split or not
- Ef/uniFB01ciently encoded with bit-string
- Shallow octree have fixed depth

Input Resolution
Accuracy
OctNet
DenseNet
VoxNet

Memory & Runtime

Experiments - 3D Classification

OctNet makes high resolution 3D convolutional networks tractable

Evaluation on Varcity dataset [3]

Experiments - 3D Orientation Estimation

Reference Commands

Conclusion
OctNet can be extended to generate high resolution 3D output [6]

Code is online: https://github.com/griegler/octnet

References