Task-driven Differentiable Grouping for Point Cloud Analysis

Zhengjie Xu*, Shuo Cheng*, Quan Vuong*

*equal contribution

Super-pixel definition:

Superpixels are an over-segmentation of an image that is formed by grouping image pixels based on low-level image properties.

Why we need this mid-level abstraction?

Advantages:

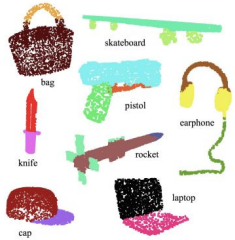
- Reduce problem complexity
- Computational efficiency

Widely used in traditional vision tasks:

- Detection
- Segmentation (i.e., in graph-cut algorithm, the super-pixel can greatly reduce the variables which need to be optimized)

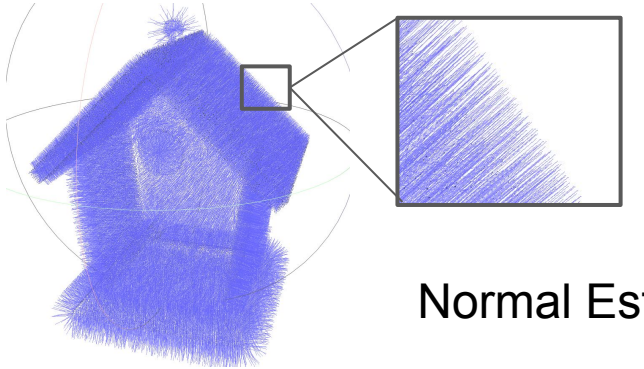
Point cloud analysis tasks:

Classification



Segmentation (object/scene)

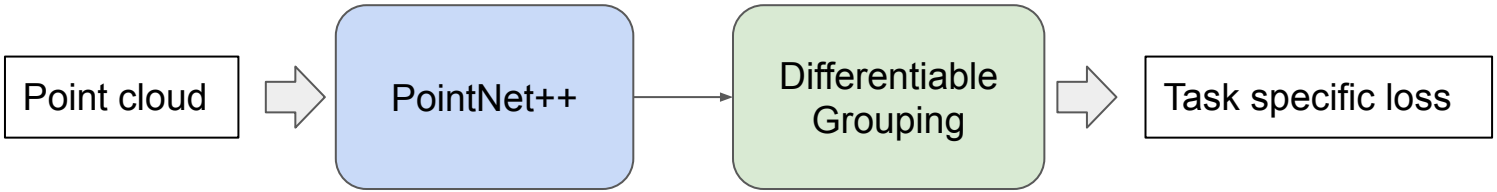
Correspondence



Normal Estimation

Task-driven Differentiable Grouping:

We implement a differentiable grouping module and combine it with off-the-shelf point cloud analysis frameworks.



The module optimize the embedding space to enforce the local feature consistency.

The learnt mid-level abstraction can be used to discover geometry primitives as well as simplify the downstream tasks.

(point cloud is not evenly distributed in the space, we hope the mid-level abstraction can somehow helps to adjust the spatial density ...)

How to make grouping differentiable?

for each iteration t in 1 to v **do**

Compute association between each pixel p and the surrounding superpixel i, $Q_{pi}^t = e^{-||F_p - S_i^{t-1}||^2}$.

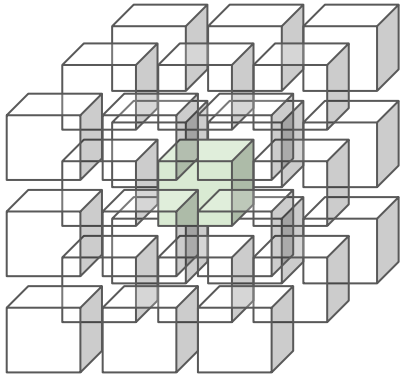
Compute new superpixel centers, $S_i^t = \frac{1}{Z_i^t} \sum_{p=1}^{n} Q_{pi}^t F_p$; $Z_i^t = \sum_p Q_{pi}^t$.

end for

Works like EM algorithm, no hard association.

Implement Details

Association in 2D

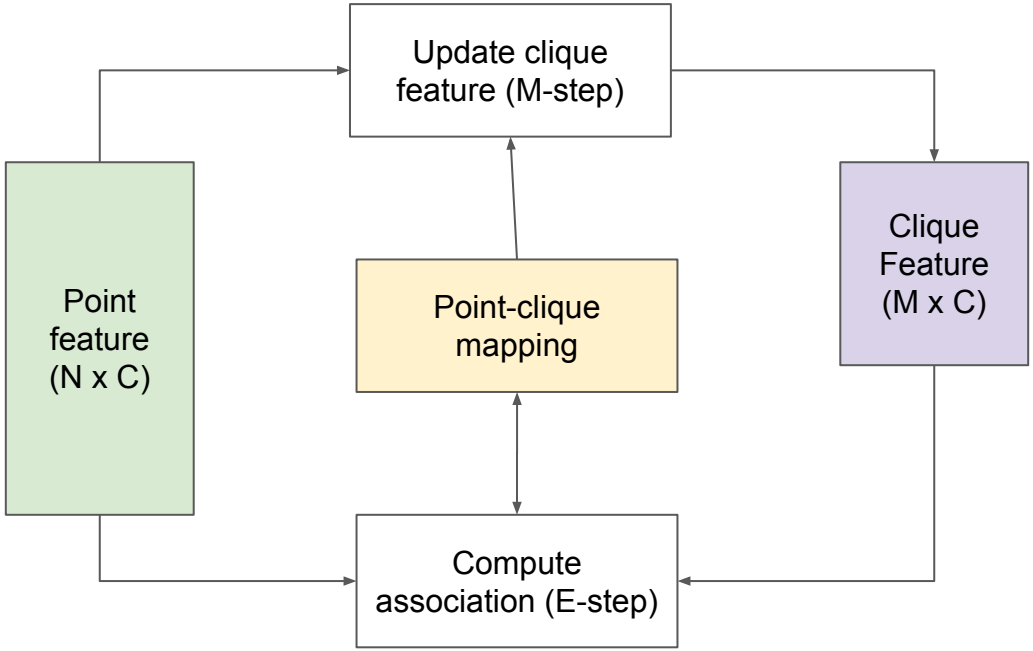


Association in 3D

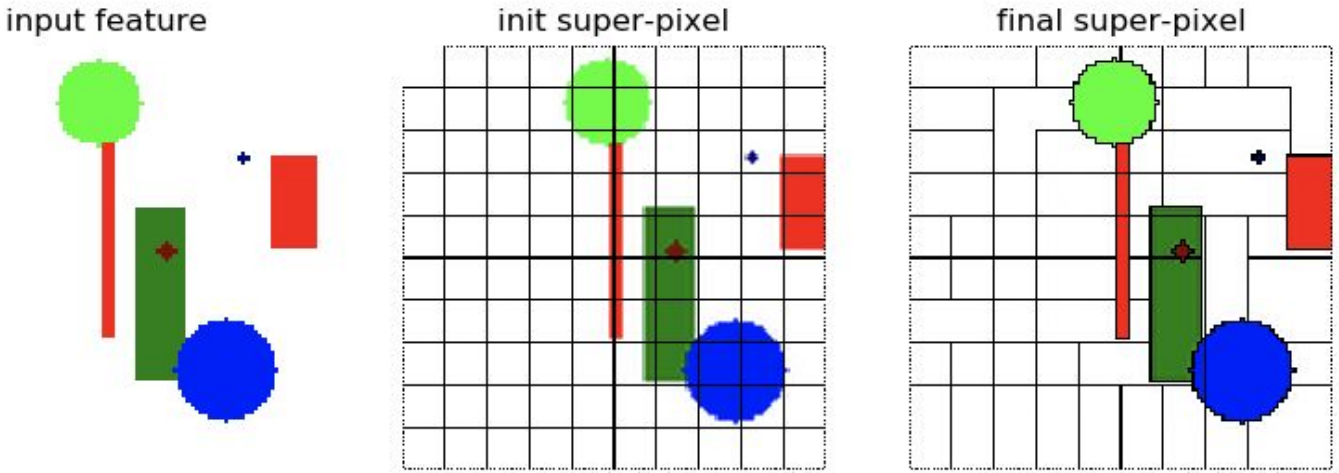
Similar to SSN, for each point, we only compute its association with surrounding 27 virtual voxel grids.

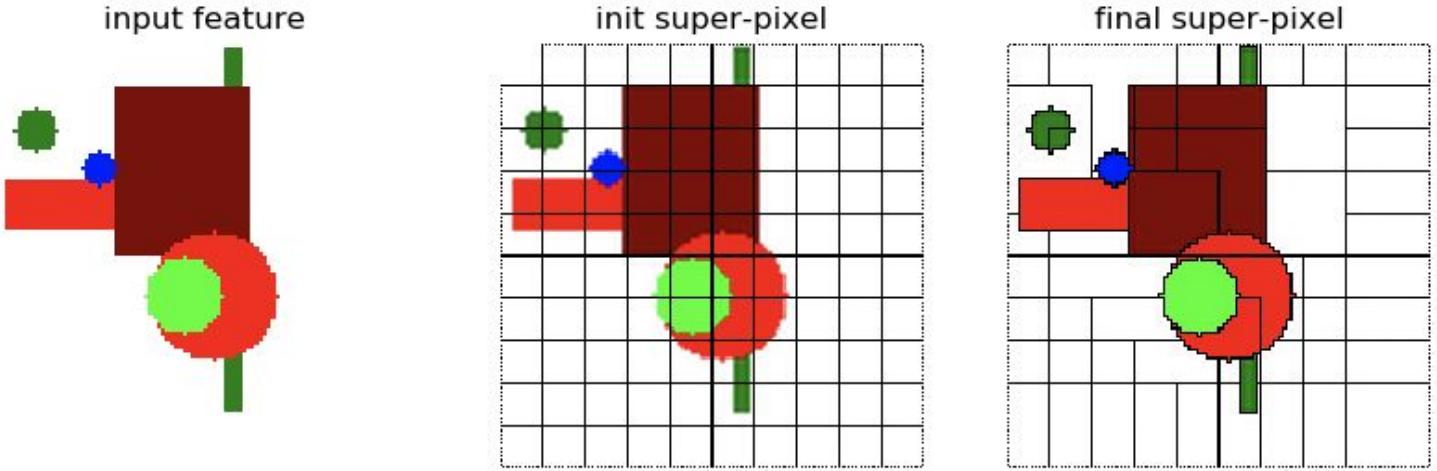
The “connection” between point and clique is predefined but the strength of each connection is learnable during training.

Indexing functions and feature collecting



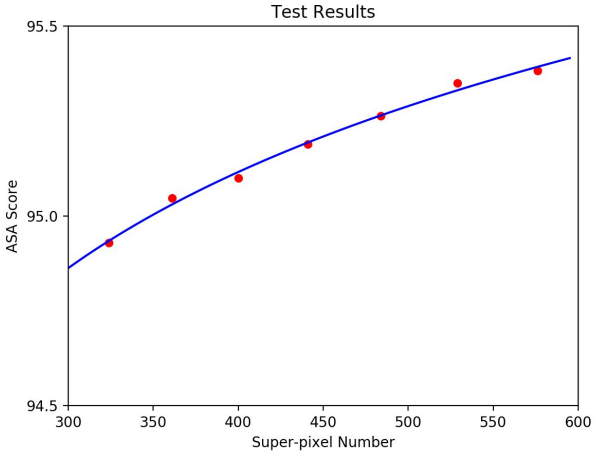
Some toy examples:

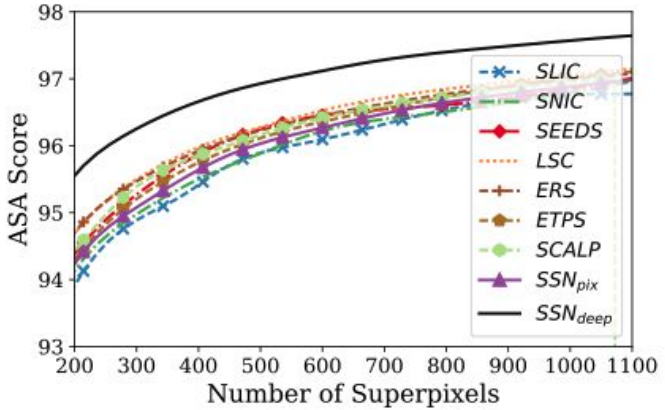



input feature
init super-pixel
final super-pixel

Re-implementation of the SSN

- Re-implemented in PyTorch
- Iterate Differential SLIC 5 times during training and 10 times during testing
- Perform additional BFS post process to merge small super pixels to the larger ones
- Get similar results comparing to the original paper

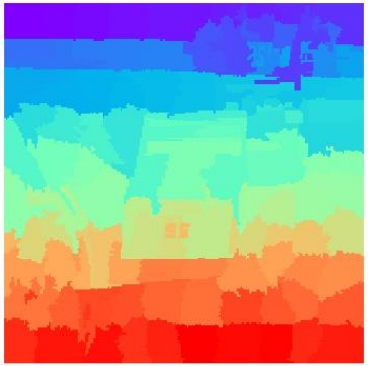




Some Qualitative Results

input img

finalized super-pixel

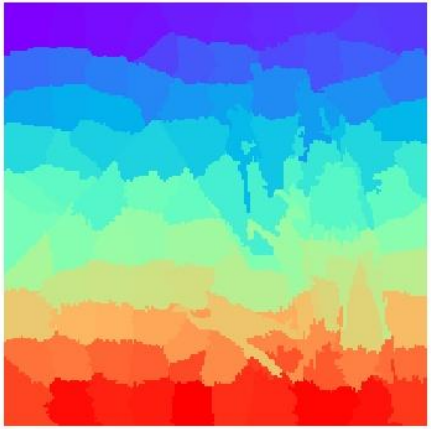


segmentation map

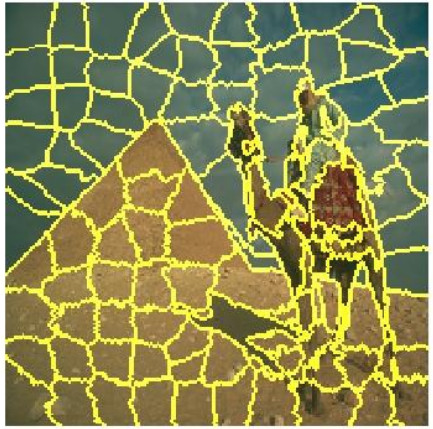
Note how the numbers here are classified in the same super-pixels

input img

finalized super-pixel

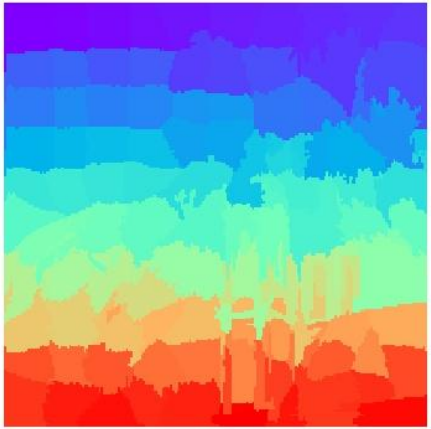


segmentation map

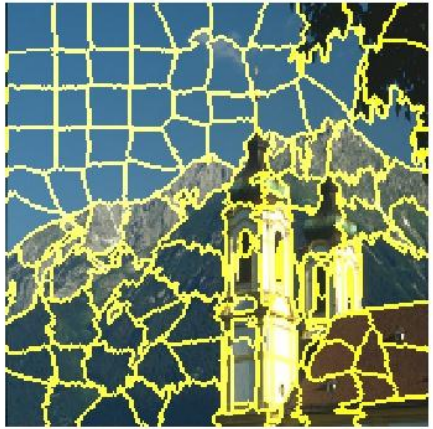


input img

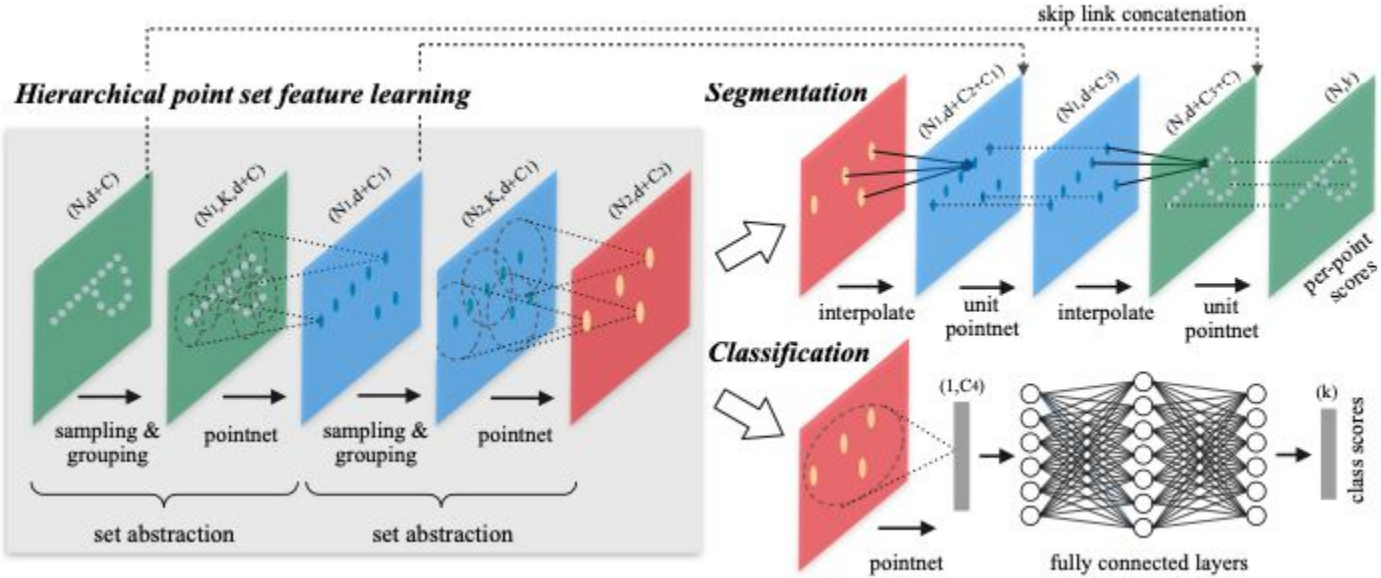
finalized super-pixel



segmentation map



Extension to PointNet++



PointNet++ Set Abstraction Layer

Given an input point cloud:

1. Iterative FPS to find centroids
2. Ball querying to find nearby points for each centroids
3. Compute features for each centroids with PointNet

=> Set Abstraction Layer: feature aggregation based on non-differentiable grouping

=> Goal: make the Set Abstraction Layer iterative and differentiable

PointNet++ Set Abstraction Layer (iterative and differentiable)

Given an input point cloud:

1. Iterative FPS to find initial centroids
2. for num_iter:
    a. Ball querying to find nearby points for each centroids
    b. For each centroid, compute distance between centroid and nearby points
    c. Compute features for each centroids with PointNet
    d. Shift centroid positions to more similar nearby points

Distance function is the exponential of negative norm of differences

=> Features of centroids and points need to have the same dimension

PointNet++ Set Abstraction Layer

	Classification Accuracy on ShapeNet40
PointNet++ baseline	0.905 (std 0.002)
PointNet++ constant dimension	0.902 (std 0.003)

Keeping the dimension of centroids the same across multiple Set Abstraction Layers have little effect on performance

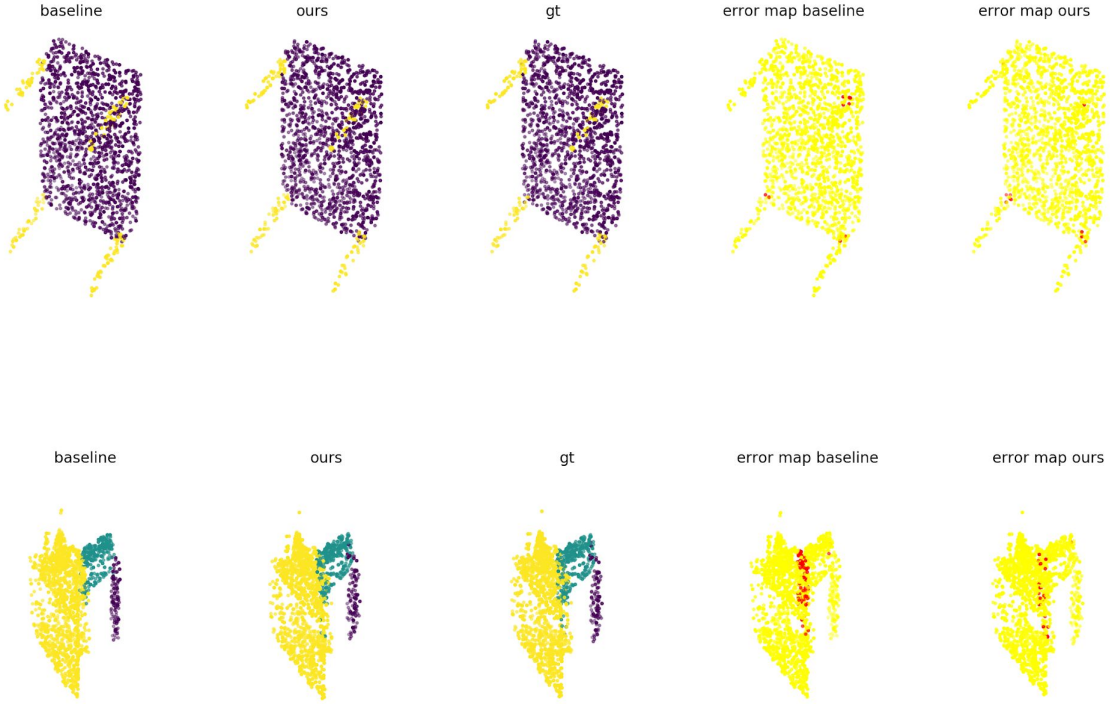
Original PointNet++ increases output dimension of each Set Abstraction Layer

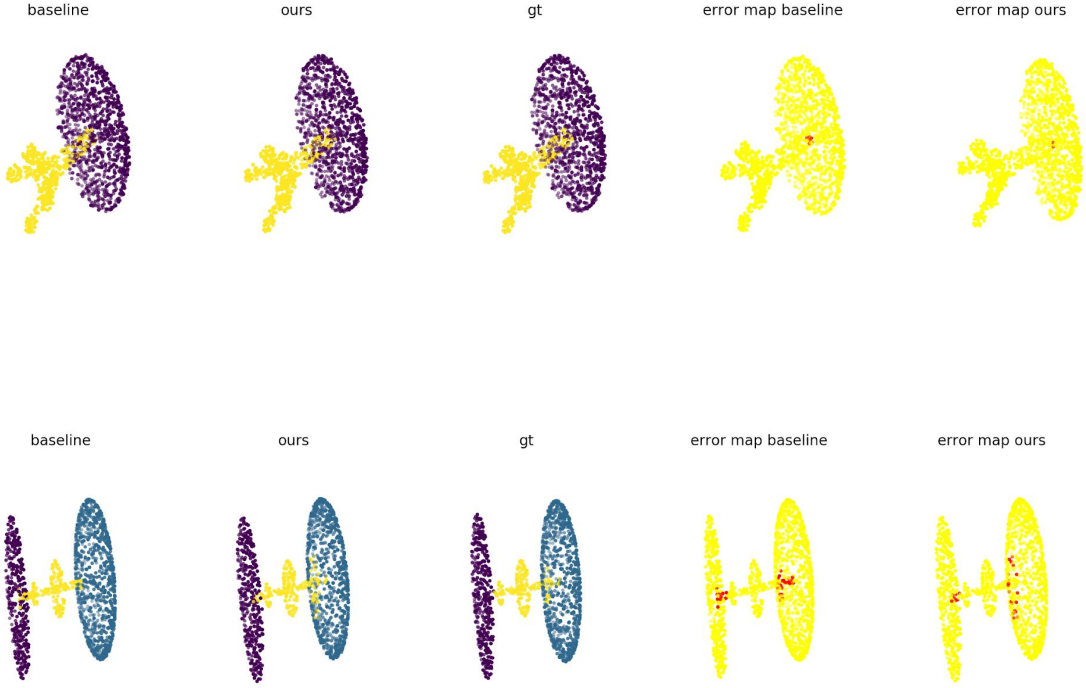
Implementation of differential grouping on Part Segmentation Task

- Perform differential grouping on the second Set Abstraction Layer (SA layer), since the first layer does not have all the features and the last one takes origin as the centroid.
- Re-querying the neighbors and compute the features after we change the centroid to a new place.
- Test different numbers of iterations for differential grouping against the baseline result of plain PointNet++

	Baseline	iter=1	iter=2	iter=3	iter=5
Seg Acc.	93.25	93.30	93.19	**93.49**	93.22
IoU	83.48	**83.82**	83.43	83.71	83.68

Qualitative Results (iteration = 3)




baseline
ours
gt
error map baseline
error map ours
baseline
ours
gt
error map baseline
error map ours