

---

# HorizonNet 코드 분석

길다영

---

2021. 07. 19

<https://github.com/sunset1995/HorizonNet> 참고.

---

# 목차

## Contents

# 1

데이터  
훈련

Preprocess.py  
Train.py

# 2

Pre-  
Processing

Preprocess.py

# 3

Estimating  
Layout  
with HorizonNet

inference.py  
/misc/post\_proc.py

# 4

Layout  
3D Viewer

Layout\_viewer.py

---

---

# 1. 데이터 훈련

Preprocess.py

Train.py

---

## 결과

1.

```
python preprocess.py --img_glob ssu_dataset/train/img/pic424_1.png --output_dir  
ssu_dataset/preprocessed --rgbonly
```

preprocess.py

이미지 조정(정렬)



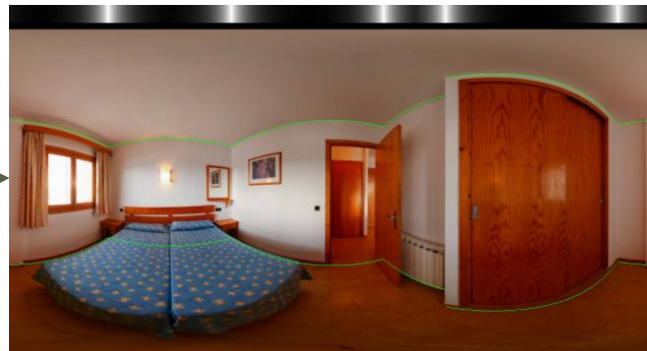
데  
이  
터

훈  
련

```
python train.py --train_root_dir ssu_dataset/train --valid_root_dir ssu_dataset/valid --  
batch_size_train 1 --num_workers 0 --id model_bs1
```

train.py

데이터 세트에 대해 학습  
Batch\_size, epochs, num\_workers 지정



```
python preprocess.py --img_glob ssu_dataset/train/img/pic424_1.png --output_dir ssu_dataset/preprocessed --rgbonly
```

1.

데이터

후련

```
51 # Process each input
52 for i_path in paths:
53     print('Processing', i_path, flush=True)
54
55     # Load and cat input images
56     img_ori = np.array(Image.open(i_path).resize((1024, 512), Image.BICUBIC))[..., :3]
57
58     # VP detection and line segment extraction
59     _, vp, _, _, panoEdge, _, _ = panoEdgeDetection(img_ori,
60                                                       qError=args.q_error,
61                                                       refineIter=args.refine_iter)
62
63     panoEdge = (panoEdge > 0)
64
65     # Align images with VP
66     i_img = rotatePanorama(img_ori / 255.0, vp[2::-1])
67     l_img = rotatePanorama(panoEdge.astype(np.float32), vp[2::-1])
```

소실점 가진 이미지를 align  
(구부러진 사진이 펴짐)

```
python train.py --train_root_dir ssu_dataset/train --valid_root_dir ssu_dataset/valid --batch_size_train 1 --num_workers 0 --id model_bs1
```

1.

데이터

관련

```
13 from model import HorizonNet, ENCODER_RESNET, ENCODER_DENSENET
14 from dataset import PanoCorBonDataset
15 from misc.utils import group_weight, adjust_learning_rate, save_model, load_trained_model
16 from inference import inference
17 from eval_general import test_general
```

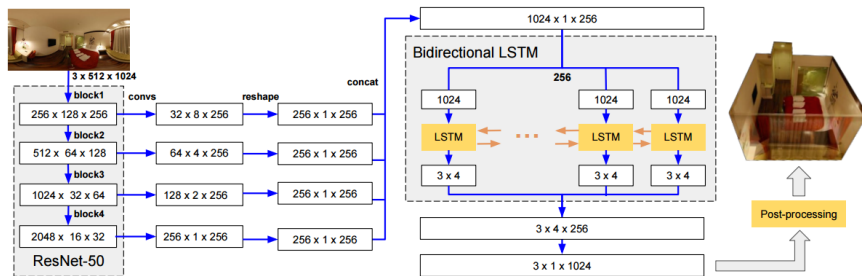
Model.py에서  
HorizonNet(ResNet50 & LSTM) 가져오기

Model.py로 이동

```
123 # Create model
124 if args.pth is not None:
125     print('Finetune model is given.')
126     print('Ignore --backbone and --no_rnn')
127     net = load_trained_model(HorizonNet, args.pth).to(device)
128 else:
129     net = HorizonNet(args.backbone, not args.no_rnn).to(device)
```

Model.py에서 가져온 HorizonNet 사용.

train.py



ResNet 50 이외의 다른 특징  
추출기 선택하여 진행해보기

RNN 존재하는 경우와  
존재하지 않는 경우 직접 비교

## CNN(ResNet50)

```

172 HorizonNet
173 ...
174 class HorizonNet(nn.Module):
175     x_mean = torch.FloatTensor(np.array([0.485, 0.456, 0.406]))[None, :, None, None]
176     x_std = torch.FloatTensor(np.array([0.229, 0.224, 0.225]))[None, :, None, None]
177
178     def __init__(self, backbone, use_rnn):
179         super(HorizonNet, self).__init__()
180         self.backbone = backbone
181         self.use_rnn = use_rnn
182         self.out_scale = 8
183         self.step_cols = 4
184         self.rnn_hidden_size = 512
185
186         # Encoder
187         if backbone.startswith('res'):
188             self.feature_extractor = Resnet(backbone, pretrained=True)
189         elif backbone.startswith('dense'):
190             self.feature_extractor = Densenet(backbone, pretrained=True)
191         else:
192             raise NotImplementedError()
193
194         # Inference channels number from each block of the encoder
195         with torch.no_grad():
196             dummy = torch.zeros(1, 3, 512, 1024)
197             c1, c2, c3, c4 = [b.shape[1] for b in self.feature_extractor(dummy)]
198             c_last = (c1*8 + c2*4 + c3*2 + c4*1) // self.out_scale
199
200         # Convert features from 4 blocks of the encoder into B x C x 1 x W'
201         self.reduce_height_module = GlobalHeightStage(c1, c2, c3, c4, self.out_scale)

```

```

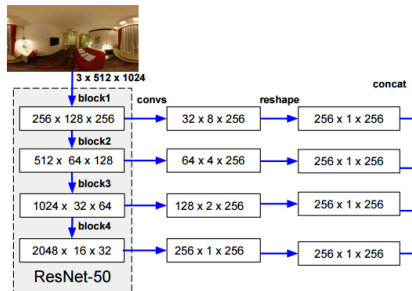
10 ENCODER_RESNET = [
11     'resnet18', 'resnet34', 'resnet50', 'resnet101', 'resnet152',
12     'resnext50_32x4d', 'resnext101_32x8d'
13 ]

```

```

53 class Resnet(nn.Module):
54     def __init__(self, backbone='resnet50', pretrained=True):
55         super(Resnet, self).__init__()
56         assert backbone in ENCODER_RESNET
57         self.encoder = getattr(models, backbone)(pretrained=pretrained)
58         del self.encoder.fc, self.encoder.avgpool
59
60     def forward(self, x):
61         features = []
62         x = self.encoder.conv1(x)
63         x = self.encoder.bn1(x)
64         x = self.encoder.relu(x)
65         x = self.encoder.maxpool(x)
66
67         x = self.encoder.layer1(x); features.append(x) # 1/4
68         x = self.encoder.layer2(x); features.append(x) # 1/8
69         x = self.encoder.layer3(x); features.append(x) # 1/16
70         x = self.encoder.layer4(x); features.append(x) # 1/32
71         return features
72
73     def list_blocks(self):
74         lst = [m for m in self.encoder.children()]
75         block0 = lst[:4]
76         block1 = lst[4:5]
77         block2 = lst[5:6]
78         block3 = lst[6:7]
79         block4 = lst[7:8]
80         return block0, block1, block2, block3, block4

```



1.

데이터

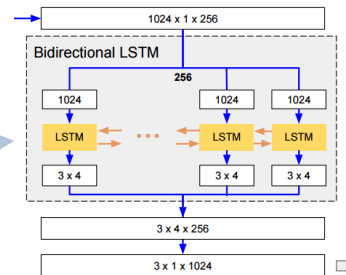
관련

## RNN(LSTM)

```

203 # 1D prediction
204 if self.use_rnn:
205     self.bi_rnn = nn.LSTM(input_size=c_last,
206                           hidden_size=self.rnn_hidden_size,
207                           num_layers=2,
208                           dropout=0.5,
209                           batch_first=False,
210                           bidirectional=True)
211
212     self.drop_out = nn.Dropout(0.5)
213     self.linear = nn.Linear(in_features=2 * self.rnn_hidden_size,
214                             out_features=3 * self.step_cols)
215     self.linear.bias.data[0*self.step_cols:1*self.step_cols].fill_(-1)
216     self.linear.bias.data[1*self.step_cols:2*self.step_cols].fill_(-0.478)
217     self.linear.bias.data[2*self.step_cols:3*self.step_cols].fill_(0.425)
218 else:
219     self.linear = nn.Sequential(
220         nn.Linear(c_last, self.rnn_hidden_size),
221         nn.ReLU(inplace=True),
222         nn.Dropout(0.5),
223         nn.Linear(self.rnn_hidden_size, 3 * self.step_cols),
224     )
225     self.linear[-1].bias.data[0*self.step_cols:1*self.step_cols].fill_(-1)
226     self.linear[-1].bias.data[1*self.step_cols:2*self.step_cols].fill_(-0.478)
227     self.linear[-1].bias.data[2*self.step_cols:3*self.step_cols].fill_(0.425)
228 self.x_mean.requires_grad = False
229 self.x_std.requires_grad = False
230 wrap_lr_pad(self)

```





## 1.

## 데이터

## 관련

```
231 def _prepare_x(self, x):
232     if self.x_mean.device != x.device:
233         self.x_mean = self.x_mean.to(x.device)
234         self.x_std = self.x_std.to(x.device)
235     return (x[:, :3] - self.x_mean) / self.x_std
236
237 def forward(self, x):
238     if x.shape[2] != 512 or x.shape[3] != 1024:
239         raise NotImplementedError()
240     x = self._prepare_x(x)
241     conv_list = self.feature_extractor(x)
242     feature = self.reduce_height_module(conv_list, x.shape[3]//self.step_cols)
243
244     # rnn
245     if self.use_rnn:
246         feature = feature.permute(2, 0, 1) # [w, b, c*h]
247         output, hidden = self.bi_rnn(feature) # [seq_len, b, num_directions * hidden_size]
248         output = self.drop_out(output)
249         output = self.linear(output) # [seq_len, b, 3 * step_cols]
250         output = output.view(output.shape[0], output.shape[1], 3, self.step_cols) # [seq_len, b, 3, step_cols]
251         output = output.permute(1, 2, 0, 3) # [b, 3, seq_len, step_cols]
252         output = output.contiguous().view(output.shape[0], 3, -1) # [b, 3, seq_len*step_cols]
253     else:
254         feature = feature.permute(0, 2, 1) # [b, w, c*h]
255         output = self.linear(feature) # [b, w, 3 * step_cols]
256         output = output.view(output.shape[0], output.shape[1], 3, self.step_cols) # [b, w, 3, step_cols]
257         output = output.permute(0, 2, 1, 3) # [b, 3, w, step_cols]
258         output = output.contiguous().view(output.shape[0], 3, -1) # [b, 3, w*step_cols]
259
260     # output.shape => B x 3 x W
261     cor = output[:, :1] # B x 1 x W
262     bon = output[:, 1:] # B x 2 x W
263
264     return bon, cor
```

1.

데

이

터

관

련

python train.py --train\_root\_dir ssu\_dataset/train --valid\_root\_dir ssu\_dataset/valid --batch\_size\_train 1 --num\_workers 0 --id model\_bs1

```

107 # Create dataloader
108 dataset_train = PanoCorBonDataset(
109     root_dir=args.train_root_dir,
110     flip=not args.no_flip, rotate=not args.no_rotate, gamma=not args.no_gamma,
111     stretch=not args.no_pano_stretch)
112 loader_train = DataLoader(dataset_train, args.batch_size_train,
113                           shuffle=True, drop_last=True,
114                           num_workers=args.num_workers,
115                           pin_memory=not args.no_cuda,
116                           worker_init_fn=lambda x: np.random.seed())

```

```

171 # Start training
172 for ith_epoch in trange(1, args.epochs + 1, desc='Epoch', unit='ep'):
173
174     # Train phase
175     net.train()
176     if args.freeze_earlier_blocks != -1:
177         b0, b1, b2, b3, b4 = net.feature_extractor.list_blocks()
178         blocks = [b0, b1, b2, b3, b4]
179         for i in range(args.freeze_earlier_blocks + 1):
180             for m in blocks[i]:
181                 m.eval()
182     iterator_train = iter(loader_train)
183     for _ in trange(len(loader_train),
184                   desc='Train ep%s' % ith_epoch, position=1):
185         # Set learning rate
186         adjust_learning_rate(optimizer, args)
187
188         args.cur_iter += 1
189         x, y_bon, y_cor = next(iterator_train)
190
191         losses = feed_forward(net, x, y_bon, y_cor)
192         for k, v in losses.items():
193             k = 'train/%s' % k
194             tb_writer.add_scalar(k, v.item(), args.cur_iter)
195         tb_writer.add_scalar('train/lr', args.running_lr, args.cur_iter)
196         loss = losses['total']
197
198     # backprop
199     optimizer.zero_grad()
200     loss.backward()
201     nn.utils.clip_grad_norm(net.parameters(), 3.0, norm_type='inf')
202     optimizer.step()

```

Epoch만큼 for문 돌림.

train.py

---

## 2. Pre-Processing

Preprocess.py

---



assets/demo.png

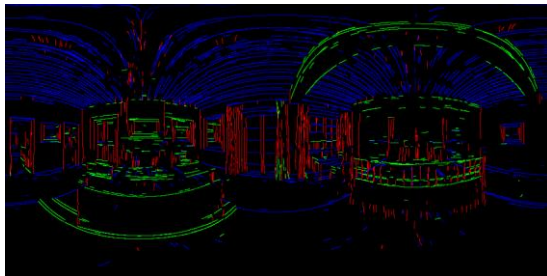
python preprocess.py --img\_glob assets/demo.png --output\_dir assets/preprocessed/

preprocess.py

demo.aligned\_rgb.png



demo.aligned\_line.png



demo\_VP.txt

demo2\_VP - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

-0.004342 -0.003421 0.999985

0.001005 0.999994 0.003426

0.999986 -0.001023 0.005268

2.

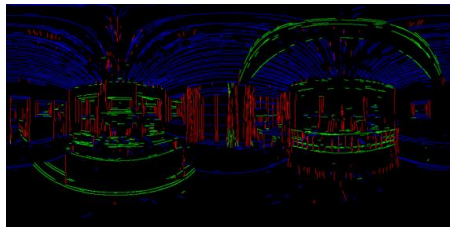
```
python preprocess.py --img_glob assets/demo.png --output_dir assets/preprocessed/
```

```

51 # Process each input
52 for i_path in paths:
53     print('Processing', i_path, flush=True)
54
55     # Load and cat input images
56     img_ori = np.array(Image.open(i_path).resize((1024, 512), Image.BICUBIC))[..., :3]
57
58     # VP detection and line segment extraction
59     _, vp, _, _, panoEdge, _, _ = panoEdgeDetection(img_ori,
60                                                     qError=args.q_error,
61                                                     refineIter=args.refine_iter)
62     panoEdge = (panoEdge > 0)
63
64     # Align images with VP
65     i_img = rotatePanorama(img_ori / 255.0, vp[2::-1])
66     l_img = rotatePanorama(panoEdge.astype(np.float32), vp[2::-1])
67
68     # Dump results
69     basename = os.path.splitext(os.path.basename(i_path))[0]
70     if args.rgbonly:
71         path = os.path.join(args.output_dir, '%s.png' % basename)
72         Image.fromarray((i_img * 255).astype(np.uint8)).save(path)
73     else:
74         path_vp = os.path.join(args.output_dir, '%s_VP.txt' % basename)
75         path_i_img = os.path.join(args.output_dir, '%s_aligned_rgb.png' % basename)
76         path_l_img = os.path.join(args.output_dir, '%s_aligned_line.png' % basename)
77
78         with open(path_vp, 'w') as f:
79             for i in range(3):
80                 f.write('%s %.6f %.6f %.6f\n' % (vp[i, 0], vp[i, 1], vp[i, 2]))
81         Image.fromarray((i_img * 255).astype(np.uint8)).save(path_i_img)
82         Image.fromarray((l_img * 255).astype(np.uint8)).save(path_l_img)

```

preprocess.py



demo2\_VP - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```

-0.004342 -0.003421 0.999985
0.001005 0.999994 0.003426
0.999986 -0.001023 0.005268

```

파노라마 사진에서  
edge와 소실점을 찾음

소실점 가진 이미지를 align  
(구부러진 사진이 펴짐)



---

### 3. Estimating Layout with HorizonNet

inference.py  
/misc/post\_proc.py

3.

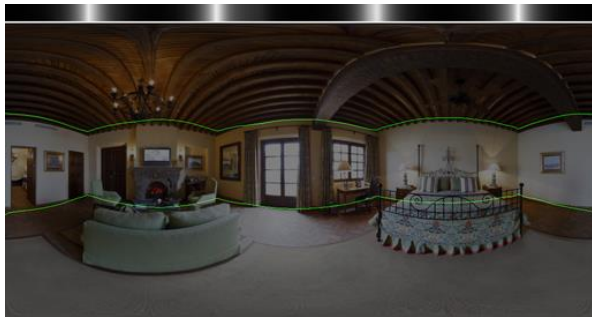
E  
s  
t  
w  
i  
i  
m  
t  
a  
h  
t  
i  
H  
n  
o  
g  
r  
i  
L  
z  
a  
o  
y  
n  
o  
N  
u  
e  
t  
t

결과

demo\_aligned\_rgb.raw.png

```
python inference.py --pth ckpt/resnet50_rnn__st3d.pth --img_glob  
assets/preprocessed/demo_aligned_rgb.png --output_dir assets/inferenced --visualize
```

inference.py



demo\_aligned\_rgb.json

demo2\_aligned\_rgb.json X

```
1 [{"z0": 50.0, "z1": -32.29869079589844,  
2 "uv": [[0.13787418603897095, 0.36454907059669495],  
3 [0.13787418603897095, 0.5889818668365479],  
4 [0.3714289665222168, 0.3713512718677521],  
5 [0.3714289665222168, 0.5841352343559265],  
6 [0.6208978891372681, 0.3664819896221161],  
7 [0.6208978891372681, 0.5875996351242065],  
8 [0.867792546749115, 0.35881996154785156],  
9 [0.867792546749115, 0.5931026935577393]]}]
```

<https://velog.io/@arittung/3D-Room-Reconstruction-Using-HorizonNet-%EC%B4%88%EA%B8%B0-%ED%99%98%EA%B2%BD-%EC%84%A4%EC%A0%95-HorizonNet-%EC%8B%A4%ED%96%89>

3.

```
python inference.py --pth ckpt/resnet50_rnn__st3d.pth --img_glob assets/preprocessed/demo_aligned_rgb.png --output_dir assets/inferenced --visualize
```

Preprocess된 이미지 불러옴

```

164 # Prepare image to processed
165 paths = sorted(glob.glob(args.img_glob))
166 if len(paths) == 0:
167     print('no images found')
168 for path in paths:
169     assert os.path.isfile(path), '%s not found' % path
170
171 # Check target directory
172 if not os.path.isdir(args.output_dir):
173     print('Output directory %s not existed. Create one.' % args.output_dir)
174     os.makedirs(args.output_dir)
175 device = torch.device('cpu' if args.no_cuda else 'cuda')
176
177 # Loaded trained model
178 net = utils.load_trained_model(HorizonNet, args.pth).to(device)
179 net.eval()

```

Trained model(pth 파일) 불러옴

Pth 파일(?)

Inference()로 넘어감.

```

181 # Inferencing
182 with torch.no_grad():
183     for i_path in tqdm(paths, desc='Inferencing'):
184         k = os.path.split(i_path)[-1][:-4]
185
186         # Load image
187         img_pil = Image.open(i_path)
188         if img_pil.size != (1024, 512):
189             img_pil = img_pil.resize((1024, 512), Image.BICUBIC)
190         img_ori = np.array(img_pil)[..., :3].transpose([2, 0, 1]).copy()
191         x = torch.FloatTensor([img_ori / 255])
192
193         # Inferencing corners
194         cor_id, z0, z1, vis_out = inference(net, x, device,
195                                           args.flip, args.rotate,
196                                           args.visualize,
197                                           args.force_cuboid,
198                                           args.min_v, args.r)
199
200         # Output result
201         with open(os.path.join(args.output_dir, k + '.json'), 'w') as f:
202             json.dump({
203                 'z0': float(z0),
204                 'z1': float(z1),
205                 'uv': [[float(u), float(v)] for u, v in cor_id],
206             }, f)
207
208         if vis_out is not None:
209             vis_path = os.path.join(args.output_dir, k + '.raw.png')
210             vh, vw = vis_out.shape[:2]
211             Image.fromarray(vis_out)\
212                 .resize((vw//2, vh//2), Image.LANCZOS)\
213                 .save(vis_path)

```

inference.py



3.

python inference.py --pth ckpt/resnet50\_rnn\_\_st3d.pth --img\_glob assets/preprocessed/demo\_aligned\_rgb.png --output\_dir assets/inferenced --visualize

2

```
65 def inference(net, x, device, flip=False, rotate=[], visualize=False,
66             force_cuboid=True, min_v=None, r=0.05):
67     ...
```

```
93 # Init floor/ceil plane
```

```
94 z0 = 50
```

```
95 _, z1 = post_proc.np_refine_by_fix_z(*y_bon_, z0)
```

논문 3.2 Recovering the Floor and Ceiling Planes

바닥, 천장 평면 초기화

```
97 # Detect wall-wall peaks
```

```
98 if min_v is None:
```

```
99     min_v = 0 if force_cuboid else 0.05
```

```
100     r = int(round(W * r / 2))
```

```
101     N = 4 if force_cuboid else None
```

```
102     xs_ = find_N_peaks(y_cor_, r=r, min_v=min_v, N=N)[0]
```

논문 3.3 Recovering Wall Planes / Figure4 (a)

Find\_N\_peaks() : prominent peak를 찾는 과정

```
104 # Generate wall-walls
```

```
105 cor, xy_cor = post_proc.gen_ww(xs_, y_bon_[0], z0, tol=abs(0.16 * z1 / 1.6), force_cuboid=force_cuboid)
```

```
106 if not force_cuboid:
```

```
107     # Check valid (for fear self-intersection)
```

```
108     xy2d = np.zeros((len(xy_cor), 2), np.float32)
```

```
109     for i in range(len(xy_cor)):
```

```
110         xy2d[i, xy_cor[i]['type']] = xy_cor[i]['val']
```

```
111         xy2d[i, xy_cor[i-1]['type']] = xy_cor[i-1]['val']
```

```
112     if not Polygon(xy2d).is_valid:
```

```
113         print(
```

```
114             'Fail to generate valid general layout!! '
```

```
115             'Generate cuboid as fallback.',
```

```
116             file=sys.stderr)
```

```
117     xs_ = find_N_peaks(y_cor_, r=r, min_v=0, N=4)[0]
```

```
118     cor, xy_cor = post_proc.gen_ww(xs_, y_bon_[0], z0, tol=abs(0.16 * z1 / 1.6), force_cuboid=True)
```

1

Gen.ww()

: cuboid, non-cuboid 함수 구분

: gen\_ww\_cuboid() → vote()

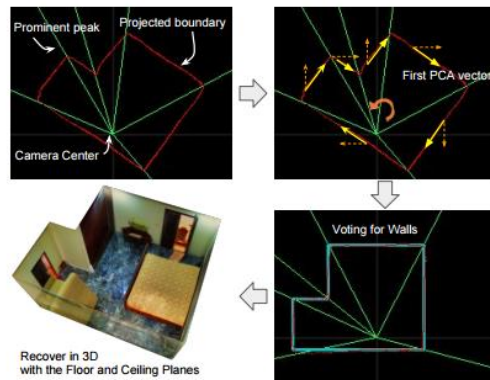
: gen\_ww\_general() → vote()

/misc/post\_proc.py로 이동

inference.py

```
21 def find_N_peaks(signal, r=29, min_v=0.05, N=None):
22     max_v = maximum_filter(signal, size=r, mode='wrap')
23     pk_loc = np.where(max_v == signal)[0]
24     pk_loc = pk_loc[signal[pk_loc] > min_v]
25     if N is not None:
26         order = np.argsort(-signal[pk_loc])
27         pk_loc = pk_loc[order[:N]]
28         pk_loc = pk_loc[np.argsort(pk_loc)]
29     return pk_loc, signal[pk_loc]
```

inference.py



(a) Depicting how we recover the wall planes from our model output.

3.

Estimate with math Non-uboid

1

3

```

337 def gen_wv(init_coors, coors, z=50, coorW=1024, coorH=512, floorW=1024, floorH=512, tol=3, force_cuboid=True):
338     gpid = get_gpid(init_coors, coorW)
339     coor = np.hstack([np.arange(coorW)[:], None], coors[:, None])
340     xy = np_coor2xy(coor, z, coorW, coorH, floorW, floorH)

    # Generate wall-wall
343 if force_cuboid:
344     xy_cor = gen_wv_cuboid(xy, gpid, tol)
345 else:
346     xy_cor = gen_wv_general(init_coors, xy, gpid, tol)
347

75 def vote(vec, tol):
76     vec = np.sort(vec)
77     n = np.arange(len(vec))[:-1]
78     n = n[:, None] - n[None, :] + 1.0
79     l = squareform(pdist(vec[:, None], 'minkowski', p=1) + 1e-9)
80
81     invalid = (n < len(vec) * 0.4) | (1 > tol)
82     if (~invalid).sum() == 0 or len(vec) < tol:
83         best_fit = np.median(vec)
84         p_score = 0
85     else:
86         l[invalid] = 1e5
87         n[invalid] = -1
88         score = n
89         max_idx = score.argmax()
90         max_row = max_idx // len(vec)
91         max_col = max_idx % len(vec)
92         assert max_col > max_row
93         best_fit = vec[max_row: max_col+1].mean()
94         p_score = (max_col - max_row + 1) / len(vec)
95
96     l1_score = np.abs(vec - best_fit).mean()
97
98     return best_fit, p_score, l1_score

```

Cuboid 경우

Non-cuboid 경우

Non-cuboid  
보완해보기

/misc/post\_proc.py

2

```

205 def gen_wv_cuboid(xy, gpid, tol):
206     xy_cor = []
207     assert len(np.unique(gpid)) == 4
208
209     # For each part separated by wall-wall peak, voting for a wall
210     for j in range(4):
211         now_x = xy[gpid == j, 0]
212         now_y = xy[gpid == j, 1]
213         new_x, x_score, x_l1 = vote(now_x, tol)
214         new_y, y_score, y_l1 = vote(now_y, tol)
215         if (x_score, -x_l1) > (y_score, -y_l1):
216             xy_cor.append({'type': 0, 'val': new_x, 'score': x_score})
217         else:
218             xy_cor.append({'type': 1, 'val': new_y, 'score': y_score})

```

For문의 개수 차이  
Cuboid : 4개  
Non-cuboid : 모서리 개수

/misc/post\_proc.py

```

241 def gen_wv_general(init_coors, xy, gpid, tol):
242     xy_cor = []
243     assert len(init_coors) == len(np.unique(gpid))
244
245     # Candidate for each part separated by wall-wall boundary
246     for j in range(len(init_coors)):
247         now_x = xy[gpid == j, 0]
248         now_y = xy[gpid == j, 1]
249         new_x, x_score, x_l1 = vote(now_x, tol)
250         new_y, y_score, y_l1 = vote(now_y, tol)
251         u0 = np_coorx2u(init_coors[(j - 1 + len(init_coors)) % len(init_coors)])
252         u1 = np_coorx2u(init_coors[j])
253         if (x_score, -x_l1) > (y_score, -y_l1):
254             xy_cor.append({'type': 0, 'val': new_x, 'score': x_score, 'action': 'ori', 'gpid': j, 'u0': u0, 'u1': u1, 'tbd': True})
255         else:
256             xy_cor.append({'type': 1, 'val': new_y, 'score': y_score, 'action': 'ori', 'gpid': j, 'u0': u0, 'u1': u1, 'tbd': True})
257

```

Occluded Corner

False Negative

벽에 투표하는 대신, 두개의 두드러진 봉우리와 두개의 벽의 위치에 따라 코너를 추가함.

2

/misc/post\_proc.py

3.

```
python inference.py --pth ckpt/resnet50_rnn__st3d.pth --img_glob assets/preprocessed/demo_aligned_rgb.png --output_dir assets/inferenced --visualize
```

E  
s  
t w  
i i  
m t  
a h  
t  
i H  
n o  
g r  
i  
L z  
a o  
y n  
o N  
u e  
t t

```
120     # Expand with btn coory
121     cor = np.hstack([cor, post_proc.infer_coory(cor[:, 1], z1 - z0, z0[:, None])])
122
123     # Collect corner position in equirectangular
124     cor_id = np.zeros((len(cor)*2, 2), np.float32)
125     for j in range(len(cor)):
126         cor_id[j*2] = cor[j, 0], cor[j, 1]
127         cor_id[j*2 + 1] = cor[j, 0], cor[j, 2]
128
129     # Normalized to [0, 1]
130     cor_id[:, 0] /= W
131     cor_id[:, 1] /= H
132
133     return cor_id, z0, z1, vis_out
```

inference.py

---

## 4. Layout 3D Viewer

Layout\_viewer.py

---

## 결과

```
python layout_viewer.py --img ssu_dataset/preprocessed/pic424_1_aligned_rgb.png --  
layout ssu_dataset/train/label_cor/pic424_1_aligned_rgb.json --ignore_ceiling --vis
```

Layout\_viewer.py



---

감사합니다.

The End

---