Deep learning-based optimal pose estimation of wave-dissipating blocks for regular supplementary works

○Yajun XU, Satoshi KANAI, Hiroaki DATE (Hokkaido University)
Tomoaki SANO, Takuya TERANISHI (Alpha Hydraulic Engineering Consultants, Co.Ltd.)

Abstract

Wave-dissipating blocks are stacked on a breakwater as armor structures to protect ports and harbors from the erosion of severe waves. Regular supplementary work is periodically required to maintain the height of the stacked blocks. This study developed a block stacking simulator where 3D block piling operation can be simulated to make the work more precise and economical. In the simulation, the piling poses of the blocks should be predicted as realistic as possible. Since the initial pose of a block before it drops has a crucial impact on the final simulation results of the block construction status, this study aims to find an optimal initialized pose of the block to fit it tightly to the existing surface of the stack of blocks. To this end, we developed a deep-learning-based optimal pose estimation method for wave-dissipating blocks for the supplementary works. The estimation performance of the proposed method is introduced in this report.

1. Introduction

A wave-dissipation block is a concrete armor structure that prevents a breakwater from erosion caused by sea waves. Due to the long-term effect of waves, regular supplementary works are required to maintain the height of the stack of blocks, as shown in **Fig. 1**. To this end, constructors must estimate the number of new blocks to be supplemented as precisely as possible. Recently, UAV-photogrammetry (UAV) and multibeam echo-sounder (MBES) could capture the as-is surface of existing blocks as a 3D point cloud. However, accurately estimating the number of supplemented blocks is still challenging.

To address the issue, so far, we developed a deep-learning-based block pose recognition from 3D point clouds of an existing block surface measured from UAV and MBES[1], predicted the locations where new blocks should be inserted on the existing blocks and simulated the block stack-up behavior using a physics engine[2]. Since the initial pose of a block before it drops has a crucial impact on the final stack-up simulation results, this study aims to find an optimal initialized pose of an additional block to fit it tightly to the existing stack of blocks using a physics engine and image-based deep-learning method.

2. Deep-learning-based estimation of the optimal initialized block pose to be inserted into existing stack of blocks

2.1 Overview of the block pose estimation

Our problem can be defined as follows: given an existing block stack as a background, find an optimal initialized pose of a new block so as to best fit to the existing stack. **Fig.2** shows the processing flow of our optimal bock pose estimation. The estimation consists of (1) creation of background block stack scene, (2) finding an optimal block pose by generate-and-test search, (3) training the network for pose estimation, and (4) prediction of the optimal initialized block pose.

2.2 Creation of background block stack scene

We used a block CAD model and a physics engine to generate the block stacking scene, as in the previous 3D block stack-up simulation [2]. The size of the simulated stacking scene is about 220m×20m. Then we create a large depth image I_d corresponding to this whole background scene.

2.3 Finding an optimal block pose by generate-and-test

Next, for generating training dataset, we should find a collection of the optimal initialized poses of an additional block so as to best fit to a given existing background block stack scene. To this end, the following generate-and-test search was adopted.

First, as shown in **Fig.3(a),** we place a small 2D window w on the depth image I_d representing the background scene. Then, we set the initial pose $(\mathbf{R},\mathbf{t})$ of a new on w before it drops so that the position $\mathbf{t}$ of the block coincide with the centroid c_w of w. To find the best fit pose to the background, we added a uniform random distribution $\delta \in [-0.5\text{m}, 0.5\text{m}]$ to each component of $\mathbf{t}$. The initial orientation $\mathbf{R}$ is also randomly perturbed.

Fig. 1 Block stacking work and the measured point clouds

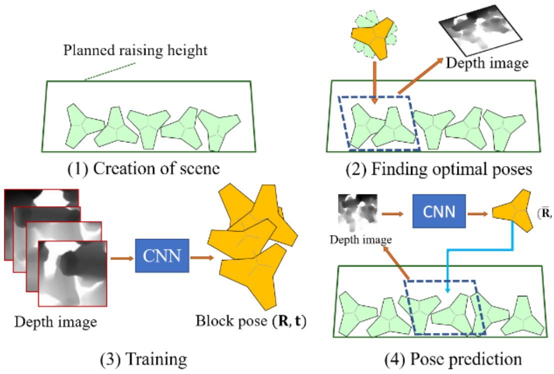
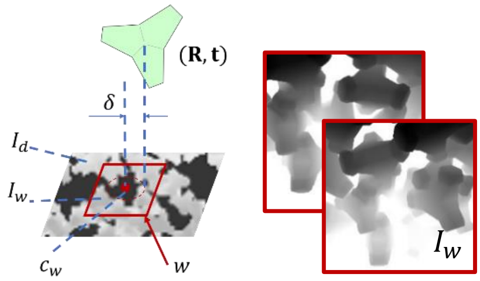


Fig. 2 The processing flow of our optimal bock pose estimation



(a) Block pose generation (b) Recorded depth images

Fig. 3 Block pose generation and samples of the recorded depth images

Finally, we drop a new block onto the background scene in the simulator, evaluate the following criteria for optimal block pose, and record the local depth image I_w cut by w.

2.4 Criteria for optimal block pose

A reasonable initial block pose needs to satisfy the two conditions.

- **Stability:** There should not be much lateral difference between the initialized pose and final stabilized pose. Therefore, the block's displacement in the horizontal plane should be less than 0.2 m (about 10% of the block size).
- **Compactness:** We want the inserted block to be as close as possible to existing background blocks. So the change in insufficient volume V between a block surface and the target height shown in Fig.4 before and after the block insertion should be small enough.

A window w slides at a fixed small interval d on the depth image I_d. At every position of w, local depth image I_w of the background scene is detected as 512×512 pixel image. Examples of I_w are shwon in **Fig.3(b).** Then we drop a new block 1000 times at a randomly selected initial pose inside I_w and evaluate the change of the insufficient volume V and block's horizontal displacement d for each time. Finally, the pose that gives the smallest change in V and satifies d<0.2m is taken as the expected optimal block pose, as shown in **Fig. 4.**

2.5 Training the network for pose estimation

In our prediction, given an input depth image I, we need to establish the correspondence f between the depth image and a reasonable pose as Eq.(1).

$$(\bar{\mathbf{R}}, \bar{\mathbf{t}}) = f(I) \tag{1}$$

where I is an input depth image, and $\bar{\mathbf{R}}$ and $\bar{\mathbf{t}}$ are the predicted rotation matrix and translation vector. To implement f as a deep-neural-network, we can use the classical feature extractor [3] or [4] followed by a several fully-connected layers, as show in **Fig .5**. To train it, we minimized the loss function,

$$\mathcal{L} = \mathcal{L}_{\mathrm{d}} + \alpha \mathcal{L}_{\mathrm{R}} \tag{2}$$

which combines displacement loss $\mathcal{L}_{\mathrm{d}}$ and rotation loss $\mathcal{L}_{\mathrm{R}}$, and α denotes a balancing constant. As $\mathcal{L}_{\mathrm{d}}$, we used the L2 loss as

$$\mathcal{L}_{\mathrm{d}}(\bar{\mathbf{t}}, \mathbf{t}) = \|\bar{\mathbf{t}} - \mathbf{t}\|_2, \tag{3}$$

where $\mathbf{t}$ is the ground truth translation vector. We took $\mathcal{L}_{\mathrm{R}}$ to be the “distance” between different pose defined by Eq.(4).

$$\mathcal{L}_{\mathrm{R}}\left(\bar{\mathbf{R}}, \mathbf{R}\right) = \min_{\mathbf{G} \in G} \cos^{-1}\left[\frac{\mathrm{tr}\left(\bar{\mathbf{R}}(\mathbf{R}\mathbf{G})^{\mathrm{T}}\right) - 1}{2}\right] \quad , \tag{4}$$

Where, $\mathbf{R}$ is the ground-truth rotation matrix, and G is the group of proper symmetries that have no effect on the static state of the object. In our problem, $\mathbf{R}$ is encoded by Euler angles.

Our network accepts a depth image I_w of size $h \times b \times 1$. After extracting a 2048-dimensional global feature $\boldsymbol{F}$, the network is divided into two branches that each pass $\boldsymbol{F}$ through a series of fully connected layers to obtain the predicted $\bar{\mathbf{t}}$ and $\bar{\mathbf{R}}$.

3. Experiment and results

We experimented with 800 depth images of 512×512 pixels on the training set and 200 images on the test set. The block type is assumed to be a clinger 6 tons. The displacement and rotation error are estimated by equation (3) and (4). We use an ADAM optimizer with initial learning rate 0.001, batch size 32. The network is trained for 700 epochs, which took about 24 hours.

The prediction results from the various feature extractors are summarized in **Table 1**. Displacement errors of less than 0.5 m were achieved, but rotational errors were not sufficiently small. The reason for this may be that it is somewhat difficult to learn the posture from the depth image only. We should consider using quaternions to express the block posture instead of Euler angles.

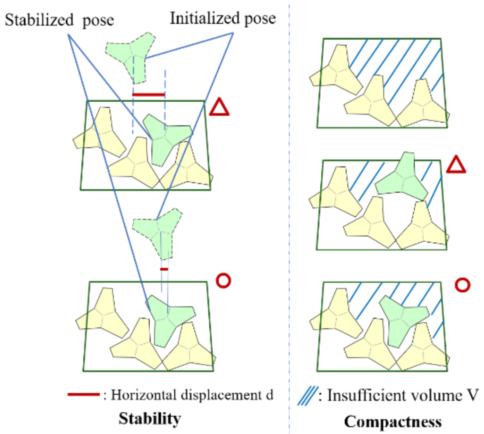


Fig. 4 Criteria for optimal block pose

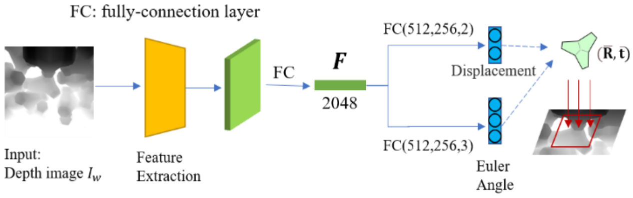


Fig. 5 Overall structure for block pose estimation

Tab. 1 Rotation and displacement errors in pose prediction.

Feature extractor	Rotation error (°)	Displacement error (m)
Transformer [4]	42.45	0.42
ResNet16 [3]	36.2	0.36
ResNet34 [3]	28.48	0.37
ResNet50 [3]	25.15	0.31

4 Summary

A deep-learning-based optimal pose estimation method for wave-dissipating blocks was proposed for the supplementary work planning, and its accuracy was verified. Although the displacement error is acceptable, the error in rotation should be more improved. The less information provided by the depth image may be a factor in poor learning. Also, we need to test more loss functions, as well as optimize the structure of the network.

Acknowledgment

This work was supported by JST SPRING, Grant Number JPMJSP2119

References

[1] Y. Xu, et al., “Recognition of wave-dissipating blocks with multiple types from 3D large-scale point clouds,” Proc. of JSPE Spring Conf, pp.53-54, (2021).

[2] Y. Xu, et al., “Prediction of 3D stacking poses of supplemental wave-dissipating blocks based on existing block poses and physics engine,” Proc. of JSPE Spring Conf, pp.156-157, (2022).

[3] K. He, et al., “Deep residual learning for image recognition,” Proc. of IEEE conference on computer vision and pattern recognition, pp. 770-778, (2016).

[4] N. Parmar, et al., “Image transformer,” Proc. of 35th international conference on machine learning, PMLR 80, pp. 4055-4064, (2018).