Semantic Metric 3D Reconstruction for Concrete Inspection

Liang Yang [1,2], Bing Li [2], Wei Li [3], Biao Jiang [2,4], and Jizhong Xiao [1,2,*]
[1] State Key Laboratory of Robotics, Shenyang Institute of Automation, UCAS
[2]CCNY Robotics Lab, City College of New York, [3] Amazon AWS AI, [4] Hostos Community College
lyang1, bli, bjiang, jxiao@ccny.cuny.edu, wayl@amazon.com *

Abstract

In this paper, we exploit the concrete surface flaw inspection through the fusion of visual positioning and semantic segmentation approach. The fused inspection result is represented by a 3D metric map with a spatial area, width, and depth information, which shows the advantage over general inspection in image space without metric info. We also relieve the human labor with an automatic labeling approach. The system is composed of three hybrid parts: visual positioning to enable pose association, crack/spalling inspection using a deep neural network (pixel level), and a 3D random field filter for fusion to achieve a global 3D metric map. To improve the infrastructure inspection, we released a new data set for concrete crack and spalling segmentation which is built on CSSC dataset [27]. To leverage the effectiveness of the large-scale SLAM aided semantic inspection, we performed three field tests and one baseline test. Experimental results show that our proposed approach significantly improves the capability of 3D metric concrete inspection via deploying visual SLAM. Furthermore, we achieve an 82.4% *MaxF1 score for crack detection and* 88.64% *MaxF1 score for spalling detection on the relabeled dataset.*

1. Introduction

The public concrete structure is affected by gradual and wide aging problem, which requires periodic inspection and evaluation in a formal routine [21], and early detection of defects is very important for long-term maintenance. However, this routine inspection has been long time performed by the human in a manual approach to carrying large and heavy equipment. According to the US Federal Highway Administration (FHWA)' latest bridge element inspection manual [2], New York Bridge Inspection Manual [21], and Tunnel Operations, Maintenance, Inspection, and Evaluation (TOMIE) Manual [3], during a routine inspection of such bridge and tunnel, it is required to identify, measure, and record information of condition state (CS). Such CS including Spall (delamination, patched area), exposed rebar, cracking, abrasion (Wear), and damage etc (see in Fig.1). Our motivation is to develop the automatic inspection using a visual camera and associate with visual positioning information to enable large-scale metric encoding.

(a) Crack (b) Spalling with exposed rebar (c) Spalling(Delamination)

Figure 1. This paper only concerns the crack(a), spalling with exposed rebar(b), and pure spalling(c) three kinds of concrete flaws. The condition states of these flaws consist of CS1 (good), CS2 (fair), CS3 (poor), and CS4 (severe) four degrees.

Visual inspection approach has been proved to be the most easy access and effective way since last century [1]. perform inspection in the pixel level due to the fact of lacking odometry information. Visual positioning (Visual Odometry or SLAM) has been heated studied since EKF mono-SLAM research [6], and later the pure optimization based SLAM with motion assumption [[14] enables the possibility of real-time processing. ORB-SLAM [19] which deploys Bag of Words and parallel threads for tracking and optimization enables on-line real-time and large-scale SLAM. Direct approach of minimizing Photometric Error [10, 8] performs pose estimation over all pixels, which is more robust in certain circumstance including image blur compared with feature approach [19]. However, there exist only one research of using SLAM to assist concrete inspection [26], and no research has been done to perform accurate semantic metric reconstruction for concrete inspection.

For visual inspection [1], deep learning based approach has been proved to be able to provide a more robust inspection performance [26, 4] compared with traditional edge detection with regression approach [12]. However, there does not exist such a publicly available dataset for concrete

*Corresponding Author

spalling and crack inspection, especially a pixel level labeled dataset for end-to-end pixel-wise segmentation training.

To achieve large-scale metric semantic inspection and measurement for the concrete structure, efficient 3D semantic reconstruction using video frame is another main issue. Authors in [15] firstly proposed probability associated occupied voxels to represent the real world in a semantic approach, and they proposed to use the conditional random field (CRF) to perform recursive fusion from frame to frame in a modeless Bayesian approach. Later authors in [28] proposed to achieve automatic semantic segmentation using a deep neural network with both RGB and depth images. Recent work by John McCormac et al [18] proposed a new 3D representation approach by introducing 3D surfels. The 3D surfels representation is proved to be much more storage efficient and dense compared with voxel representation approach. More recent research on using a recurrent neural network (RNN) to perform large-scale 3D semantic fusion also shows promising performance [24].

However, the following challenges still exist and are urgently needed to be solved: 1) high-quality dataset of concrete visual spalling and crack defects; 2) a semantic segmentation approach to support efficient pixel-level detection, with metric information of flaw areas such as width, depth, and area size. 3) 3D semantic reconstruction and detection updating from continuous frames. In this paper, we present a large-scale semantic 3D reconstruction method for concrete structure spalling and crack detection with metric measurement, which is composed of three parts: SLAM as positioning association, deep neural network as defects segmentation, and conditional filter approach for sequence fusion as 3D semantic reconstruction.

2. Method

In this section, we discuss the framework of 3D semantic reconstruction system for concrete spalling and crack metric measurement. It is illustrated in Fig.2, where the 3D metric concrete inspection system is composed of three parts, which are visual SLAM system of deploying visual positioning, a deep neural network for inspection, and a Bayesian filter for 3D semantic fusion. The visual SLAM is performing through a front-end estimation and back-end optimization pipe-line to provide real-time positioning, then the pose is used to perform 3D data association and registration for large-scale metric estimation. We also discuss the data preparation and tools in detail, and we further release our source code and data [1]. Finally, we discuss the 3D semantic fusion in a filter approach to obtain sequence-based metric reconstruction.

[1] https://github.com/ccny-ros-pkg/inspectionNet_Segmentation

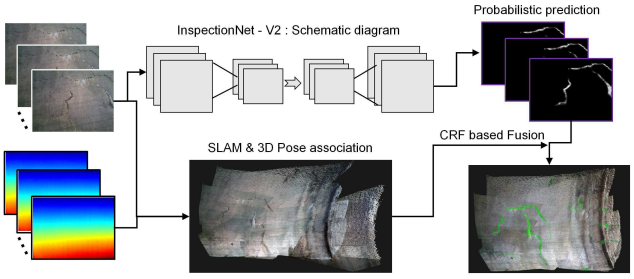


Figure 2. 3D metric inspection system framework. The input is RGB-D sequence, then visual SLAM and deep inspection are performed to achieve data association. Finally, a Bayesian filter is used to obtain 3D metric map fusion.

2.1. SLAM and Data Association

We choose RGB-D cameras to perform our visual positioning, which is inspired by our previous research on RGB-D based visual odometry [7] and local/global closing proposed in ORB-SLAM2 [19]. For each newly obtained frames (RGB image and depth image), the visual SLAM system is to perform pose estimation. The transformation between two consecutive frames as $R \in SO(3)$ as (where $SO(3)$ special Lie rotation group), and $t \in R^3$ denotes the translation in the world coordinate system. Each step motion of two consecutive frames is achieved using ICP for in a feature cloud domain. Given two consecutive frames I_p and I_q with corresponding features $\mathrm{F}_{I_p}, \mathrm{F}_{I_q}$, the transformation can be represented as:

$$\{R,t\} =_{R,t} \sum_{i\in\{1,\dots,N\}} L_\rho(\mathrm{F}_{I_p}(i) - \pi(||R\cdot \mathrm{F}_{I_q}(i)+t)||^2_{\sum}) \tag{1}$$

where denotes a linear regression process toward minimal, $L_\rho(\cdot)$ is the Huber loss cost function, and $||_{\sum}$ denotes the covariance weighted sum toward a robust convergence. Then, the pose T^F of each current frame is obtained through a cumulative approach. A co-visibility graph is also constructed locally and globally to perform local and global optimization to maintain scale and decrease long-term drift.

Pre-association between the raw images and the pose T^F as $S^P = \{F_{RGBD}, F_{Depth}, T^F\}$ is the main issue of semantic SLAM, where F_{RGB}, F_{Depth} denotes the raw image an depth image. However, a simple depth registration of the point cloud does not meet the needs of our metric measurement. We deploy CRF which is in the same approach as described in [15] for octree-voxels fusion and [13] for surfels fusion, to perform sequence map fusion. In this paper, since our purpose is to obtain the metric information of the defects area, we test both representation approaches.

2.2. InspectionNet for Concrete Structure Inspection

To associate SLAM pose, pixel-level concrete defects inspection using the deep neural network is proposed in our system. Unlike the region based detection, which was proposed in work [26], we aim to provide a pixel-level segmentation with 3D reconstruction toward the area, width, and length measurement as requested by N.Y DOT [21]. To offer a possible answer to such challenge, we re-labeled the CSSC dataset [26] and discussed in detail in Section.3, and we also proposed a new minor edge oriented network in Section.4.

2.3. CRF as Fusion

For 3D semantic reconstruction M, each surfel (or voxel) $\Im$ is designed to save the distribution probability $P_C = \{P_{c_i}|ç_i, i = 1, ..., Ç\}$, where Ç denotes the number of classes which is trained in the network.

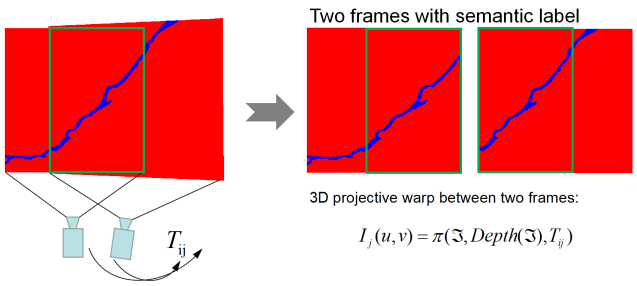


Figure 3. Illustration of merging of two frames with the semantic label. The blue rectangle region denotes the overlapping region of two frames, and two frames are illustrated on the right side.

The general approach of fusing two frames is illustrated in Fig.3, where each $\Im$ denotes a 3D unit of the map which initialized with the left single image at the right side. For each image frame F_{RGB}, the semantic prediction is performed via using the InspectionNet, that is, each pixel $I(u, v)$ in F_{RGB} will be labeled overall class labels with probabilistic distribution $P(I(u, v) = ç_i|Ç)$, where (u, v) is the coordinate in an image frame F_{RGB}. The prediction of each unit is independent of other frames [16, 15] which is just conditional distribution without generative measurement model requirement called CRF. For each unit $\Im$, we initialize with uniform possibility over each class as $P(\Im)$. Then, the next frame overlapping region, we perform a projection via deploying a general homogeneous transformation:

$$I_j(u, v) = \pi(\Im, Depth(\Im), T_{ij}) \tag{2}$$

Where $Depth(\Im)$ the depth of unit $\Im$ in the current image frame, and T_{ij} is the transformation from the last frame to current frame. Then, the corresponding pixel probabilistic prediction of $\Im$ in the current frame as $P(I_j = ç_i|F_{RGB}(j))$, and we can update the probabilistic distribution following a recursive Bayesian update procedure:

$$P(\Im = ç_i|F) = P(I_i(u, v))P(I_j) \tag{3}$$

The posterior update is carried over all units, which must be activated at the current frame. It can also be seen from Fig.3, the 3D space with surfel (or voxel) description also needs partition which can be found in [15].

3. Spalling/Cracking Data Annotation and Segmentation Model

3.1. Dataset Annotation

The dataset to be annotated is provided by Liang Yang et al [26], called Concrete Structure Spalling and Crack (CSSC) database. However, the spalling image in CSSC was initially proposed to do region-based classification using fine-tuned VGGNET [23]. This paper performed further annotation on the dataset to do semantic segmentation. We defined the following guidelines to be the key for high-quality annotation: (1) only concrete spalling and cracking meaningful regions should be annotated; (2) annotation only perform at targeted spalling and cracking region, other regions should be annotated as background. (3) the spalling region should be annotated with polygons; (4) crack region should be detailed annotated in pixel level, especially unclear cracking. These guidelines enable us to label carefully with spalling and cracking.

(1) *Spalling annotation*: CSSC dataset is only labeled with eroded steel region (as illustrated in Fig.4.b). In this paper, we introduce to use Labelme to do spalling region labeling. We name the spalling region as 'spalling', and each annotator is asked to follow the definition provided by civil engineers to label the corresponding spalling region. The annotation only performs on such region which can be named as spalling, where the boundaries should be able to provide a clear comparison. Thus, multiple polygons exist for spalling in one image, and we name the other regions as background (Fig.4.c). Finally, we further process the labels to generate expected ground truth images.

(2) *Crack annotation* Crack region tends to more scale variant and with low contrast, and we further checked the CSSC dataset which already provides the part of labeled images. Annotators are asked to label the minor crack regions over all the images with the semantic name tag. Besides, we should pay attention that if a crack region is blurred, the visible crack regions should all be annotated.

(3) *Data augmentation* To increase the network robustness and desired invariance for both orientation and illumination, especially when only limited data is given for training. For our concrete inspection case, rotation and illumi-

(a) original image (b) CSSC Label (c) Our semantic annotation

Figure 4. (a) concrete spalling is the flaw that concrete breaks down into small spalls from the concrete body. (2) The label from CSSC dataset. (3) Our newly labeled image using a closed polygon to describe the spalling part.

nation are main issues affecting the accuracy. According to [26] that illumination and image blur affect the detection accuracy a lot. We perform the following augmentation techniques: 1) flipping and rotation, an approach to increase rotation robustness; 2) gamma transformation [11] with $X^{'} = 255 \cdot (1 + X/255)^{\gamma}$, where X denotes the image, γ is designed to correct the intensity and we increase the contrast of image to increase the illuminance robustness; 3) sub-sampling, to increase the robustness with input scale.

3.2. Related Segmentation Networks

The InspectionNet is motivated by HED and U-net, and these two deep neural networks are end-to-end fully pixel-level segmentation for edge segmentation (InspectionNet is illustrated in Fig.5).

3.2.1 HED Network

HED improved VGGNET with the following aspects: 1) it connects side output from $conv1\_2$, $conv2\_2$, $conv3\_3$, $conv4\_3$, and $conv5\_3$ to the last convolutional layer. 2) it trimmed $5th$ pooling layer as well as all the following fully-connected layers. For all the 5 side outputs S, where each layer have to perform deconvolution to do upsampling, their corresponding classification weights are $w^s = \{w_1^s, ..., w_5^s\}$. Thus, the objective function is a linear fusion which is defined as:

$$\mathcal{L}(W, w) = \sum_{j=1}^{5} \alpha_j \ell^j(W, w^j) \tag{4}$$

where $\alpha. = 0.2$, W denotes the kernel parameters, $\ell^j(W, w^j$ denotes the image-level loss function for side-outputs. The loss function in [25] in this case is defined as an evaluation over all pixels of ground truth compared to predicted output, especially, the paper defines a class-balanced cross-entropy loss function for each side-output.

$$\begin{aligned} \ell^j(W, w^j) = &-\beta \sum_{i \in Y_+} logPs(y_i = 1|X; W, w^j) \\ &-(1-\beta) \sum_{i \in Y_-} logPs(y_i = 0|X; W, w^j) \end{aligned} \tag{5}$$

where $y_i = 0, 1$ is edge information and background information respectively, Y_- and Y_+ denote edge and non-edge label in ground truth image, $\beta = |Y_-|/|Y|$, and $Ps(\cdot)$ is the sigmoid activation output on side-outputs.

3.2.2 U-Net

U-net [22] was proposed to perform end-to-end segmentation without fully connected layer. It consists of 4 groups of convolutional layer with max-pooling, 4 groups of convolutional layers with 'up-sampling', and a final group of convolutional layers with $1 * 1$ convolutional kernels. Each group has two convolutional layers with a 3×3 kernel and Relu. Besides, each convolutional layer performs convolution without padding, thus leads to a final 388×388 output if given 572×572 input.

U-net introduces a pixel-wise soft-max to perform loss calculation over predicted feature map with given ground truth. Given image set $X = \{X_m | X_m = \{x_i^m, i = 1, ..., |X_m|\}, m = 1, ..., M\}$, the soft-max in [22] is defined as

$$p_k(x_i^m) = exp(a_k(x_i^m))/(\sum_{k'=1}^{K} exp(a_{k'}(x_i^m))) \tag{6}$$

where $a_k(x_i^m)$ denotes the activation at feature channel k at pixel position x_i^m, K denotes the number of clusters, $p_k(x_i^m)$ denotes the approximate maximum-function. The loss based on the cross entropy is defined as

$$E = \sum_{x_i^m \in X_m} w^U(x) log(p_{(k)(x_i^m)}(x_i^m)) \tag{7}$$

where $(k) \in \{1, ..., K\}$ is the label of each pixel, and w^U is the corresponding importance weight.

Remark: The best of HED is an end-to-end edge detection, and it trimmed the traditional fully connected layers, which thus increased the time performance and decreased the model size. For U-net, it is end-to-end pixel level prediction by combining spatial and contextual information [5]. U-net has a total of 19 convolutional layers with a cross-entropy based loss function. The network obtains up-sampling with convolutional kernel to perform accurate prediction of region-based prediction compared to HED.

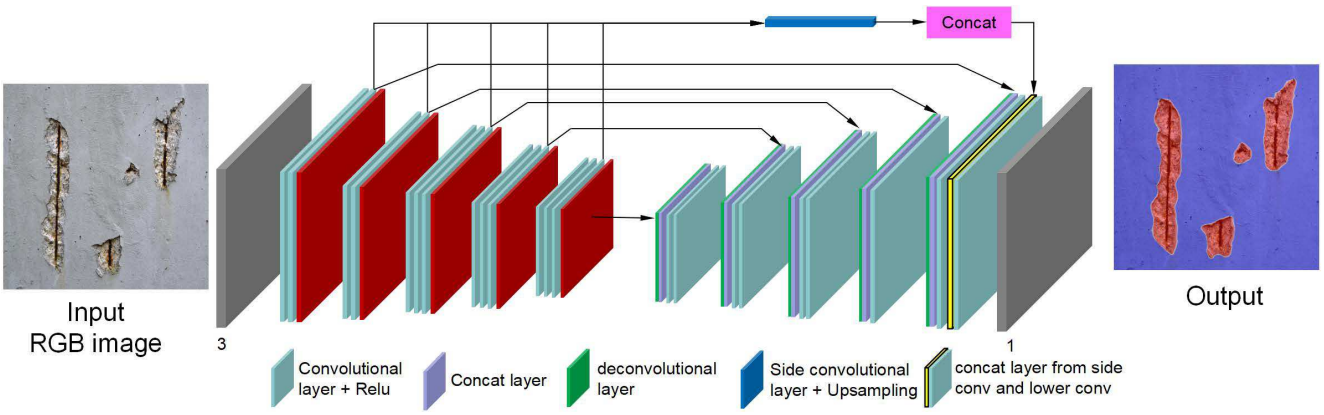


Figure 5. The InspectionNet model, the model has a total 37 layers which is motivated by U-net and HED net.

4. InspectionNet Model

4.1. Crack/Spalling Model Overview

Our model is a highly hybrid model which consists of two components, illustrated in Fig.5. The two parts are jointly trained end-to-end to optimize our semantic spalling and crack segmentation quality by employing edge information. The network combines the merits of U-net [22] and HED [25], U-net structure act as an end-to-end pixel-wise prediction and HED side-outputs performs edge extract intermediate layers to enable further feature exploiting.

We consider using VGGNet to fit the U-net structure as the first component, with a total 27 convolutional layers (inherit from Vgg-16 for left side) [20]. The original VGGNet is also trimmed following HED by only obtaining the first five group of convolutional layers. For each deconvolutional layer, the number of features is doubled with a concatenation from previous encoder layers. We deploy a padding for each convolutional layer, thus guarantees a complete pixel-wise mapping from input to output. Each convolutional layers is followed by an element-wise rectified linear non-linearity (ReLU) $max(0, x)$, and the max-pooling has a stride 2 and 2×2 window size. The deconvolutional layers with a stride of 2 and kernel size of 3 to densify the sparse activations obtained by performing a convolution-like operation with learned filters.

The second component of our network is that we introduced the side-output convolutional (total 10 layers) as HED of concatenation with the final convolutional layer to introduce edge feature estimation. It is illustrated in Fig.5 that the blue diagram is the side-layer, which performs pixel-wise estimation by using convolution with a 1×1 size kernel and bilinear interpolation. The bilinear interpolation may be performed several times due to the different size of the original side-output, and the convolutional layer is also followed by a Relu to remove negative values. We also deploy a loss calculation for each side-output to perform side optimization as proposed in [25]. Then, the five edge features will be concatenated with a final convolutional layer to perform the pixel-wise prediction.

We highly take advantage of Vgg-16 pre-trained model and transfer the entire low-level features to do prediction. The model performs a complete same size convolution with padding to guarantee a complete mapping from input 2D dimension to output prediction. Our model introduces a totally 27 convolutional layers to perform feature exploration, which is much deeper compared to HED and U-net. Furthermore, the side-output prediction involves a better estimation of contour compared to U-net. We also take full advantage HED net pre-trained weighted of side-output convolutional layers.

4.2. Loss Design and Training

Segmentation seeks the high pixel-wise overlapping between the prediction and ground truth. The cross-entropy based on the pixel-wise estimation probability $Ps(\cdot)$ (as discussed in Equ. (2) and Equ. (3)) is commonly used as the loss function, where the probability of $Ps(\cdot))$ is usually a weighted probability as discussed in Equ.5 and Equ.7. In our paper, the spalling and cracking do not commonly happen in one image and we only care about the cracking or spalling region, thus we adjust the objective function from Equ.1 from [25] as

$$\mathcal{L}(W, w) = \sum_{j=1}^{6} \alpha_j^{'} \ell^j(W, w^j) \tag{8}$$

where the weight$\alpha^{'}_{\cdot}$ is adjusted to shift higher weight to the final convolutional layer output.

The training is a two-step procedure. Firstly, we re-train the HED in the same way as proposed in [25] using Berkeley Segmentation Dataset and Benchmark (BSDS 500) [17] dataset which has 200 training, 100 validation, and 200

testing images. Then, we use the side-convolutional layers weight to initialize the model's side-convolutional layers and use Vgg-16 weight to initialize the weight of our model first five groups' convolutional layers. The decoding layers (the right side layers) are randomly initialized. For all the layers, the parameters are allowed to be able to update.

4.3. Evaluation

We aim at developing measurements to quantify algorithm performance on our dataset, and also performs an evaluation of the performance of proposed network using such measurements. Since the spalling and cracking region detection behaves as a region-based segmentation, we compare with the following perspectives: 1) F1 score: $F1 = 2*(precision*recall)/(precision+recall)$; 2) average precision to indicate the average pixel-wise accuracy of the evaluation: $AP = Truepositive/(Truepositives+Falsepositive)$. We also evaluate the intersection over union (IOU) and enable the visualization of the cross-entropy loss as well the training precision.

5. Experimental Evaluation

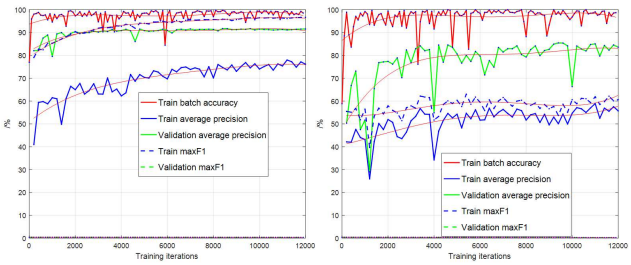
To provide a comparative and quantitative measurement of our system, we begin by performing model training and validation performance comparison with the current most successful algorithm to provide a basic baseline for peer researcher. We also demonstrate the performance of our large semantic segmentation aid reconstruction based on SLAM. For all the algorithm, we run on a GPU server with GTX 1080, and a Core I7 computer with 32G memory. Field test demo of semantic 3D reconstruction with inspection is as shown in demo video [2].

5.1. Model Training Analysis

Dataset Based on the CSSC dataset, in which 278 spalling images with exposed rebar labeling and 954 crack image with 104 labeled images, we further expand spalling images to 298. For training purpose, we have a total 298 spalling image with pixel level labeling, and 522 crack images with pixel level labeling. In addition to the original labeled images, we further cropped the large image size to a maximum of $1,600 \times 1,100$, and we also perform flipping to augment the images. Then, we get a total of $4,473$ images for the crack model, where $3,147$ images for training, 498 for cross-validation, and 828 for testing. For spalling detection, we have 627 for training, 90 for cross-validation, and 177 for testing. Inspection performance of both spalling and cracking model is measured using batch concurrent accuracy, average precision, and max F1 score [9].

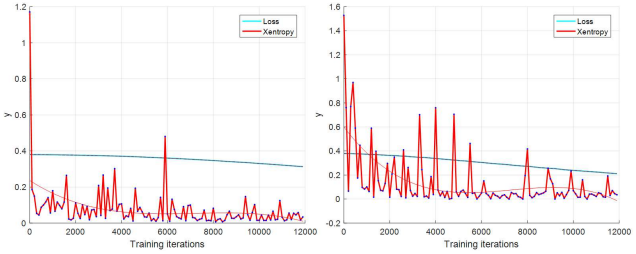
Crack Model

[2] https://youtu.be/juOwwwROPN0



(a)Training with partial dataset (b)Training with complete dataset

Figure 6. We provide a comparative training on InspectionNet using (a) partial dataset with 104 crack images. (b) the complete dataset with 522 crack images, and batch concurrent accuracy, average accuracy, and max F1 score are compared for the two cases.



(a)Training with partial dataset (b)Training with complete dataset

Figure 7. The comparison of training loss and entropy of InspectionNet with (a) partial dataset and (b) the complete dataset.

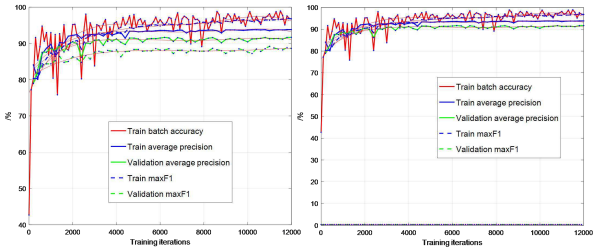
For crack inspection, we performed several comparisons with FCN-8s and Unet, where the Unet is a modified of using VGG-16 as initialization (we released the whole network to perform automatic updating with transfer learning). we found that FCN-8s is not able to detect the crack, and it is illustrated in Table.1 that VGG-Unet (since we use Vgg-16 as initialization) can achieve 76.67% average accuracy, and a 58.89% maxF1 score. To validate the performance of performing training on different scale dataset, we also trained using the partial training dataset which is provided by CSSC origin dataset with 104 images to build the training and testing dataset. It is illustrated in Fig.6 that the raw concurrent batch accuracy can reaches 95% within $1,000$ iterations. However, as one can see in the graph that the InspectionNet can only reach 83.58% average precision of complete dataset compared to 91.5% of performing training on whole 522 images generated the dataset. For loos and entropy, as shown in Fig.7, the partial training dataset can lead to faster convergence. In this graph, it also shows that the loss of performing complete dataset using InspectionNet is harder to converge. However, we validate in the field test that the partial model has a much higher False Positive ratio compared with complete dataset trained model.

Spalling Training

Spalling training is also executed in $12,000$ steps, and

Table 1. Comparison Of InspectionNet In the perspective of Accuracy Performance. E_MaxF1 is evaluation MaxF1 score, E_AP is evaluation average precision, T_MaxF1 is training MaxF1 score, T_AP is training average precision, T_BAP is training concurrent precision, AF is the average frequency.

	InspectionNet		VGG-Unet		FCN-8s	
	Crack	Spalling	Crack	Spalling	Crack	Spalling
E_ MaxF1	82.40	88.64	76.76	88.72	-	88.68
E_ AP	83.59	91.69	80.96	91.71	-	91.54
T_ MaxF1	60.60	96.69	58.89	96.79	-	96.64
T _ AP	55.70	93.81	55.70	93.94	-	93.78
T_ BAP	98.00	95.00	94.00	95.00	-	94.5
AF	8.02	6.53	6.48	6.78	-	6.48



(a)VGG - Unet (b)InspectionNet

Figure 8. Comparison between (a) VGG-Unet model and (b) our InpectionNet. The perspectives of batch concurrent accuracy, average accuracy and max F1 score are compared.

also in a transfer learning approach of using the VGG-16 model parameter and HED to initialize the InspectionNet. To provide a baseline for the spalling detection, we compared the performance of InspectionNet with the FCN-8s net and HED network. The comparative result of average precision, the max F1 score is represented in Table.1. As one can see that all models can achieve an average precision over 90% and maxF1 score 88.64% within $4,000$ iterations, and the average precision and MaxF1 of InspectionNet for spalling detection are almost the same compared with other two models. For long-term performance, we can see that the InspectionNet is more stable compared with VGG-Unet since our InspectionNet has a higher order feature information to assist residue passing (see in Fig.8).

5.2. Dataset Test and Field Test

The evaluation of the visual inspection system is performed in two steps. First, we test the detection performance on the test dataset and evaluate the average accuracy. In the second step, we perform field tests in several places located in Manhattan, New York, with semantic reconstruction. In the field tests, we consider both normal illumination and low illumination situation to perform inspection and 3D reconstruction.

The performance of performing detection on relabeled CSSC dataset is illustrated in Fig.9. In this figure, D_T1 denotes test on the dataset, where $D_T1:(1)\ (5)$ are spalling detection result and $D_T1:(5)\ (10)$ are crack detection result. For crack detection on the dataset, we have an average precision of 76.41%. The average precision of spalling detection is 87.9319%. F_T1 and F_T2 denotes two sets of test. $F_T1:(1)\ (10)$ illustrate that the InspectionNet can perform detection very well on field data, where the minor cracks can be easily segmented out. $F_T2:(1\ 5)$ indicate the segmentation of original image for defects. $F_T2:(6\ 10)$ denotes the detection with dark illumination. $Comp$ denotes the segmentation comparison between InspectionNet $((1)\ (5))$ and VGG-Unet $(6)\ (10)$, where we can see InspectionNet has a better performance with the minor crack inspection.

3D Metric Semantic Registration

We perform two tests which are represented in Fig.10. The 3D reconstruction is performed by coupling the image frames with $pose$ (achieve through SLAM) and $time$, where the frames are key-frames for SLAM. Then, the InspectionNet detects the region of defects. Thus we can register to 3D space with the semantic labeled image. However, a pure voxels based registration without fusion does not able to provide clear result for civil engineers (see in Fig.11.(a)), and thus this paper introduces the filter based fusion approach to perform 3D fusion as illustrated in Fig.11.(b). We further show two detailed reconstruction in Fig.11.(c) and (d). We can see in Fig.11 that the fusion approach can provide higher level of details than a pure voxels registration. Besides, we also performed two more field test as illustrated in Fig.10, where Fig.10.(a) and Fig.10.(c) are the real scenario overlayed with color, and Fig.10.(b) and Fig.10.(d) are the semantic 3D map.

6. CONCLUSION

In this paper, a semantic metric 3D reconstruction based concrete inspection system is developed for the civil engi-

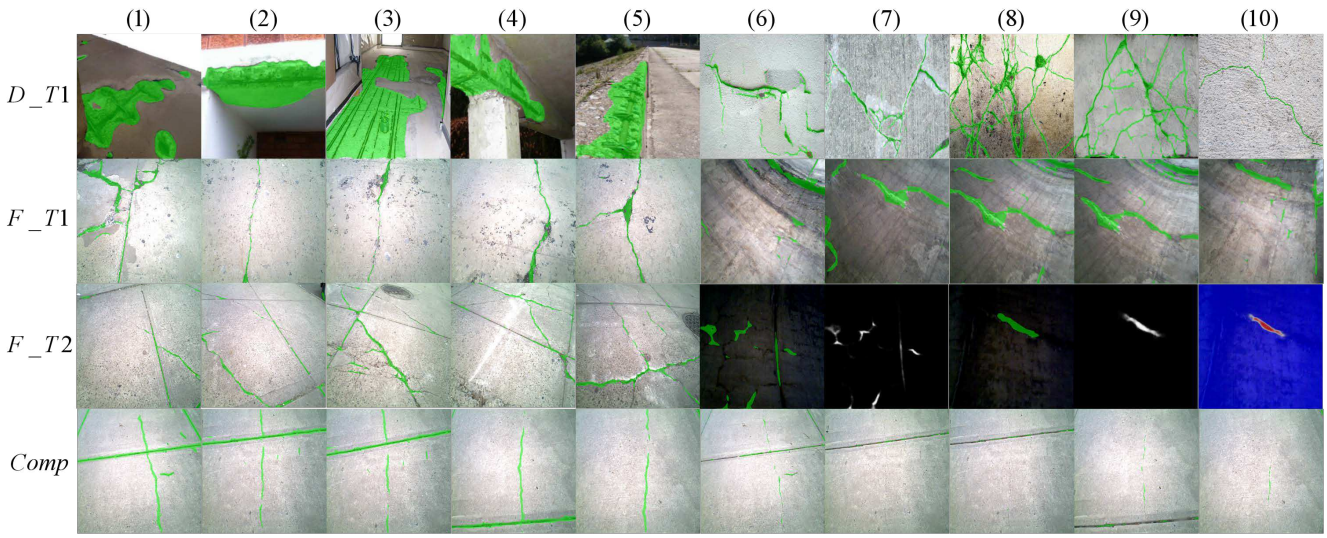


Figure 9. An illustration of detection results on relabeled dataset and field collected data. The green color denotes the defects region of the original image, the red color is used to highlight the defect region, and the white and black image is the original output of the InspectionNet.

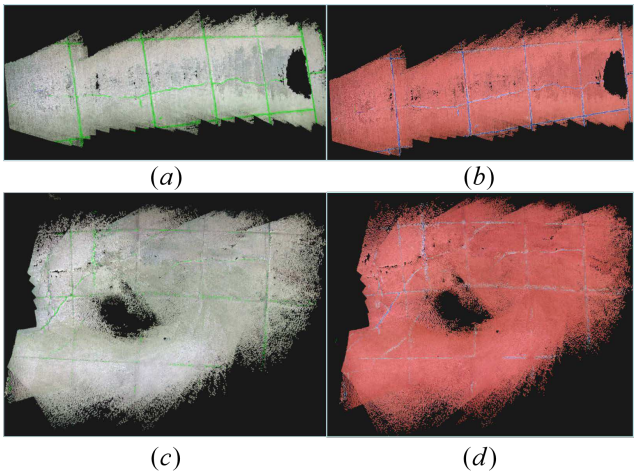


Figure 10. 3D semantic reconstruction generated by our system, which performs fusion with association from SLAM and deep neural network. (a) and (b) show the same area, where (a) is the reconstruction based on original data and (b) is the semantic map. (c) and (d) also show the same area with the same meaning as (a) and (b).

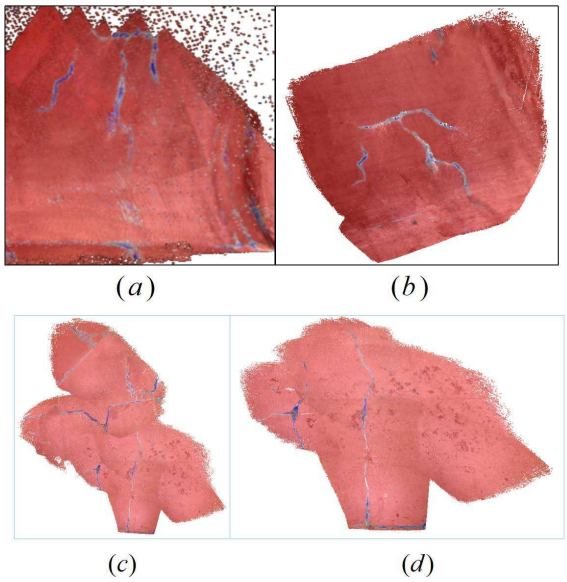


Figure 11. The illustration and comparison of filter based fusion and pure registration using voxel. (a) is pure 3d registration, (b) is filter based fusion, (c) and (d) are 3D fusion results.

neering application. A state-of-the-art dataset with pixel-level labeling and an InspectionNet network were designed for semantic segmentation. Furthermore, we bridge the gap between perception and localization using CRF as 3D fusing to perform 3D reconstruction, where the detected results can be registered in 3D model to provide metric information for concrete structure condition assessment. To evaluate the system, we executed both field tests and dataset test. The system can achieve as high as over 80% accuracy with both crack and spalling inspection for 3D information retrieve.

7. Acknowledgement

This work is partially supported by University Transportation Center on INSpecting and Preserving Infrastructure through Robotic Exploration (INSPIRE Center) with USA Federal HighWay Administration (FHWA) grant FAIN 69A3551747126.

References

[1] R. Adhikari, O. Moselhi, and A. Bagchi. Image-based retrieval of concrete crack properties for bridge inspection. *Automation in construction*, 39:180–194, 2014.

[2] F. H. Administration. Specification for the national bridge inventory bridge elements. 2014.

[3] F. H. Administration. Tunnel operations, maintenance, inspection, and evaluation (tomie) manual. 2015.

[4] Y.-J. Cha, W. Choi, and O. Büyüköztürk. Deep learning-based crack damage detection using convolutional neural networks. *Computer-Aided Civil and Infrastructure Engineering*, 32(5):361–378, 2017.

[5] P. F. Christ, M. E. A. Elshaer, F. Ettlinger, S. Tatavarty, M. Bickel, P. Bilic, M. Rempfler, M. Armbruster, F. Hofmann, M. DAnastasi, et al. Automatic liver and lesion segmentation in ct using cascaded fully convolutional neural networks and 3d conditional random fields. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 415–423. Springer, 2016.

[6] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1052–1067, 2007.

[7] I. Dryanovski, R. G. Valenti, and J. Xiao. Fast visual odometry and mapping from rgb-d data. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 2305–2310. IEEE, 2013.

[8] J. Engel, T. Schöps, and D. Cremers. Lsd-slam: Large-scale direct monocular slam. In *European Conference on Computer Vision*, pages 834–849. Springer, 2014.

[9] M. Everingham and J. Winn. The pascal visual object classes challenge 2012 (voc2012) development kit. *Pattern Analysis, Statistical Modelling and Computational Learning, Tech. Rep*, 2011.

[10] C. Forster, M. Pizzoli, and D. Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 15–22. IEEE, 2014.

[11] S.-C. Huang, F.-C. Cheng, and Y.-S. Chiu. Efficient contrast enhancement using adaptive gamma correction with weighting distribution. *IEEE Transactions on Image Processing*, 22(3):1032–1041, 2013.

[12] M. R. Jahanshahi and S. F. Masri. Adaptive vision-based crack detection using 3d scene reconstruction for condition assessment of structures. *Automation in Construction*, 22:567–576, 2012.

[13] M. Keller, D. Lefloch, M. Lambers, S. Izadi, T. Weyrich, and A. Kolb. Real-time 3d reconstruction in dynamic scenes using point-based fusion. In *3D Vision-3DV 2013, 2013 International Conference on*, pages 1–8. IEEE, 2013.

[14] G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 225–234. IEEE, 2007.

[15] A. Kundu, Y. Li, F. Dellaert, F. Li, and J. M. Rehg. Joint semantic segmentation and 3d reconstruction from monocular video. In *European Conference on Computer Vision*, pages 703–718. Springer, 2014.

[16] B. Limketkai, D. Fox, and L. Liao. Crf-filters: Discriminative particle filters for sequential state estimation. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 3142–3147. IEEE, 2007.

[17] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[18] J. McCormac, A. Handa, A. Davison, and S. Leutenegger. Semanticfusion: Dense 3d semantic mapping with convolutional neural networks. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 4628–4635. IEEE, 2017.

[19] R. Mur-Artal and J. D. Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[20] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1520–1528, 2015.

[21] N. Y. D. of Transportation. Bridge inspection manual. January, 2016.

[22] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[23] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[24] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017.

[25] S. Xie and Z. Tu. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1395–1403, 2015.

[26] L. Yang, B. Li, W. Li, Z. Liu, G. Yang, and J. Xiao. Deep concrete inspection using unmanned aerial vehicle towards cssc database. In *IEEE/RSJ International Conference on Intelligent Robots and Systems 2017*. IEEE, 2017.

[27] L. Yang, B. Li, W. Li, Z. Liu, G. Yang, and J. Xiao. A robotic system towards concrete structure spalling and crack database. In *Robotics and Biomimetics (ROBIO), 2017 IEEE International Conference on*, pages 1276–1281. IEEE, 2017.

[28] C. Zhao, L. Sun, B. Shuai, P. Purkait, and R. Stolkin. Dense rgb-d semantic mapping with pixel-voxel neural network. *arXiv preprint arXiv:1710.00132*, 2017.