This item was submitted to [Loughborough's Research Repository](#) by the author.
Items in Figshare are protected by copyright, with all rights reserved, unless otherwise indicated.

Automated sewer pipe defect tracking in CCTV videos based on defect detection and metric learning

PLEASE CITE THE PUBLISHED VERSION

<https://doi.org/10.1016/j.autcon.2020.103438>

PUBLISHER

Elsevier

VERSION

AM (Accepted Manuscript)

PUBLISHER STATEMENT

This paper was accepted for publication in the journal Automation in Construction and the definitive published version is available at <https://doi.org/10.1016/j.autcon.2020.103438>

LICENCE

CC BY-NC-ND 4.0

REPOSITORY RECORD

Wang, Mingzhu, Srinath Shiv Kumar, and Jack C.P. Cheng. 2020. "Automated Sewer Pipe Defect Tracking in CCTV Videos Based on Defect Detection and Metric Learning". Loughborough University.
<https://hdl.handle.net/2134/13186790.v1>.

Automated sewer pipe defect tracking in CCTV videos based on defect detection and metric learning

Mingzhu Wang^{a*}, Srinath Shiv Kumar^b, Jack C.P. Cheng^c

^a School of Architecture, Building and Civil Engineering, Loughborough University, Loughborough, United Kingdom

^b Artificial Intelligence Engineer, SewerAI, United States

^c Department of Civil and Environmental Engineering, the Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong

Abstract

Computer vision techniques are widely studied for automating the interpretation of sewer pipe inspection videos, yet previous studies mainly focus on defect detection and segmentation of individual images, which cannot identify if the defect is the same one across consecutive video frames (i.e. track the defect). Nevertheless, the number of unique defects in the video is required for evaluating the pipe condition. This paper proposes a framework for tracking multiple sewer defects in CCTV videos based on defect detection and metric learning. First, a deep learning -based defect detection model and a metric learning model is developed and trained respectively using with our sewer datasets. Then, using the detections and their features from the trained models as inputs, the tracking module predicts tracks by Kalman filter and associates tracks based on defect motion, appearance features, and defect types. Our experiments demonstrate the framework is able to track sewer defects in CCTV videos with a decent IDF1 score of 57.4%. We notice that tracking performance can be influenced by the detection accuracy and configurations of the metric learning module. By analyzing the tracking results based on different weights of the distance metrics, we find that assigning larger weights to appearance and defect class distance metrics tends to increase IDF1 score, while larger motion distance weight may degrade tracking accuracy. The proposed framework contributes by tracking multiple sewer defects, which can assist with counting unique defects in inspection videos.

Keywords

Defect tracking, sewer defect, sewer inspection, defect detection, defect counting, computer vision, deep learning

1. Introduction

Sewer pipe systems are essential components of underground utilities - the “blood vessels” in cities to provide urban services. In many countries, most sewer pipe systems are suffering from various defects due to their long service history of 30 to more than 100 years [1]. Those defects can speed up the pipe deterioration and lead to various issues, such as sanitary sewer overflows (SSOs), inflow and infiltration as well as sinkholes. It is estimated that there are 23,000 to 75,000 SSOs each year in the US, which can be caused by blockages in pipes, leading to water contamination and threats to public health [2]. The ASCE graded the condition of wastewater (sewer) infrastructure in the US as “D+” in 2017 and estimated that \$271 billion is required for wastewater systems in the next 25 years, with more than half of which spent for the operation & maintenance (O&M) sector [3]. The severe conditions of sewer systems and the large financial requirement for O&M indicate the high importance and necessity of efficient sewer inspection and assessment, such as to schedule maintenance activities and allocate resources appropriately.

Vision-based sewer inspection techniques such as closed-circuit television (CCTV) are commonly utilized to investigate interior conditions of sewer pipes by taking videos and images. During the inspection, the inspectors need to manually detect existing defects and record the type, location and number of each defect according to related codes, manuals or standards. After inspection, the captured videos may need to be reviewed again to confirm defect information and evaluate sewer conditions. Although different standards are adopted in different countries and regions, such as the Pipeline Assessment Certification Program (PACP) [4] in the US and Hong Kong Conduit Condition Evaluation Codes (HKCCEC) [5] in Hong Kong, the main information required for evaluating sewer condition are similar, typically including the defect type, location or distribution in the video frame, distance along the pipeline, the number of each type of defects as well as the defect severity score. Nevertheless, the manual interpretation process to acquire those information requires huge amount of time and efforts, and the results can be inconsistent due to different understanding of the defect condition from different inspectors, i.e. the conditions can be over-estimated or under-estimated.

To address the limitations of manual interpretation, computer vision techniques can be leveraged to understand objects in the inspection videos automatically and assist with identifying defects for inspectors, which is expected to greatly reduce the workloads and increase the assessment efficiency. More importantly, if historical inspection videos are available, computer vision methods can be utilized to review the inspection videos of each pipeline recorded in previous years, the results of which can be used to model the deterioration process and predict future conditions. Corresponding to the common tasks required for assessing sewer condition, previous computer vision -based studies mainly focus on automatically identifying defect type [6], detecting relative defect locations with bounding boxes in video frames [7,8] as well as segmenting the shape of defects to analyze geometric characteristics [9]. The aforementioned research has attempted to identify defect type and relative location in each given image through defect detection models. However, inspectors also need to obtain the number of each type of defect in the video to evaluate the sewer pipe condition. Based on the defect detection alone, we can only detect the defect in each static

frame but cannot know if the defect detected in the previous frame is the same one as that in the current frame, i.e. it is not easy to obtain the number of each unique defect in the video through object detection. One possible solution to obtaining the defect number based on the detection model is to detect the defect in each frame and then manually identify if the defect is the same or not across multiple consecutive frames. Such solution requires extra time and obviously is not efficient. Alternatively, object tracking is a technique to track objects continuously and assign a unique identification (ID) number to each object in a video, which is a potential method for tracking different sewer defects in the video and help counting the defect number in the end. Despite the tracking process is more complex than detection, it is worthwhile to explore its feasibility considering that defect tracking may alleviate the time and efforts required by manually counting the defect number.

Although sewer defects are not moving objects, the inspection equipment with the camera is moving along the pipeline and taking videos during inspection. Consequently, sewer defects in the captured video can be regarded as moving objects. To obtain the unique defect number, this paper proposes to track each defect in the video and assign a unique ID number to each defect. The problem of tracking various sewer defects in the video is similar to multiple object tracking (MOT) such as pedestrian tracking and vehicle tracking, but not totally the same. Tracking different sewer defects is more like multiple-class multiple-object tracking since more than one type of defects need to be tracked, while most MOT methods only track objects of the same class. Therefore, in addition to object motion and appearance feature that are commonly used in MOT approaches, we also incorporate the object class information, i.e. the defect type, into the tracking algorithm, which is not considered in most state-of-the-art MOT methods.

In this study, we develop a framework for tracking sewer defects in CCTV videos, which consists of three main modules: defect detection, metric learning and defect tracking. A deep learning -based detection model is developed and trained to obtain a high accuracy for detecting fractures, tree roots and sewer laterals. Meanwhile, a metric learning model is trained to extract discriminative features of defects such as to re-identify the defects in the video. In the defect tracking module, a defect tracking method is proposed based on three distance metrics such as to track multiple sewer defects more accurately. As the outputs from the detection model and metric learning model are used as the inputs for the tracking process, we also investigate the influence of the performance and configurations of detection and metric learning on the final tracking results.

The remainder of the paper is structured as the following: Section 2 reviews related studies on vision-based sewer defect detection and assessment, as well as multiple object tracking. Section 3 introduces the proposed framework for tracking sewer defects and the contained three main modules, followed by evaluation metrics. Section 4 demonstrates the experiments for validating the framework and studying the influence of different modules. Section 5 concludes the article.

2. Related works

2.1. Vision-based sewer defect detection and assessment

Vision-based studies for automated interpretation of sewer pipe defects from inspection images, such as CCTV or SSET images can be traced back to the year of 1994 [10] and are

evolving all along the way till now. The majority of previous studies rely on conventional computer vision methods, especially image processing techniques, for various tasks across different stages for the automated inspection, such as pre-processing, detection and segmentation, feature extraction and classification [10]. With the availability of dataset and computing resources in recent years, deep learning has attracted the attention of researchers to address limitations of previous studies. It should be noted that the definition of vision tasks in the field of deep learning is slightly different from that those used in conventional computer vision. The following sub-section reviews studies based on conventional computer vision and deep learning approaches.

2.1.1. Conventional computer vision and machine learning based methods

Before the tremendous adoption of deep learning techniques, conventional computer vision methods based on image processing are mostly utilized. Edge detection methods (e.g. Sobel operator[11,12], Canny edge detector [13,14], Laplacian of Gaussians (LoG) [15,16]) are utilized to detect object edges or changes among defects and surrounding environment. Thresholding techniques are adopted to determine regions of interest in an image by simply setting a static threshold specific for certain cases, making it not generic to dynamic and complex scenarios [15–17]. Therefore, Otsu's method is widely utilized while iterative approaches are used for addressing the dynamic challenges [18–21]. Morphological operation is another main type of techniques commonly applied for removing image noise and holes [19–21] as well as segmenting defects in images. Morphological method works by encoding the features of objects in the input image using a structuring element and using the features for segmentation [22]. For example, different methods for defect segmentation were proposed using structuring elements (SEs) with increasing sizes or different orientations. Yang and Su [11,23] also proposed Morphological Segmentation based on Edge Detection (MSSED) for identifying defects such as cracks, joints and holes. However, the performance of morphological methods can be greatly influenced by the parameters of the applied SEs. Additionally, shape fitting methods, e.g. least squares estimation and Hough transformation, are used to fit the pipe joint shape to detect broken pipe or deformation [24], as well as to extract laterals [25] or pipe joints [26].

In addition, feature extraction is one of the basic steps to obtain useful features for image classification or object detection, one main stream of methods for which is using hand-crafted or designed descriptors. For example, Histogram of Oriented Gradients (HOG) is used to extract features for detecting tree root intrusions [20] while Scale Invariant Feature Transform (SIFT) is adopted to detect changes between two consecutive frames such as to discover the existence of defects [27]. Maximally stable external regions (MSER) is applied to extract text features from images [28]. The GIST feature descriptor is also applied and sometimes achieves better performance than HOG and SIFT [29,30]. Nevertheless, extracting features based on designed descriptors requires large amount of time and efforts to design descriptors, which may not be generic for other datasets.

Classification of defects refers to recognizing defect type in a given image. Conventional classification methods are based on heuristic rules as well as machine learning classifiers [10]. As different defects usually own different geometric characteristics, many studies based on heuristic rules classify defects by setting thresholds on various geometric properties, e.g. object

area [12,15,17], length and width [31], as well as other geometry features. In addition, methods of comparing or relating multiple images are also utilized as one of the steps for classification, such as calculating distance between color histograms [32], and measuring difference among two images [33]. Interestingly, other data related to the inspection images or videos such as the displayed text on the screen [28] and the optical flow information [21,34] are also employed for digging more useful features to assist with classifying defects.

As for the machine learning approaches, multi-layer perceptron (MLP) has been commonly utilized for classifying multiple sewer defects [35–37]. Other techniques such as fuzzy logic [38,39] and finite element analysis [40] are also combined with MLP for defect classification. Support vector machines (SVMs) is another typical classifier used in several studies, such as detecting tree roots [20], segmenting defects [41], and classifying the camera view type [21]. Yang and Su [37] also proposed ensemble SVMs for multi-defect classification. In addition, Random Forests (RFs) have been commonly employed for identifying the existence of defects [29,30] as well as classifying defect type [42]. Previous studies presented that detection results can be improved by combining RFs with SVMs. Researchers also proposed a hierarchical approach based on RFs by first identifying whether there are defects in the image and then recognizing the defect type if there is any [42]. Meanwhile, fuzzy classifier is another type of machine learning method applied. Several studies proposed neuro-fuzzy (NF) approaches, in which the input features are fuzzified using three membership functions [43,44]. Chae et al. [38] applied a set of fuzzy rules for pipe condition rating after obtaining the attributes of defects.

2.1.2. Deep learning-based methods

According to the sewer pipe condition assessment standards, such as PACP in the United States, there are different tasks required for evaluating sewer conditions, such as identifying defect type (e.g. crack, tree root, and deposit), defect location and/or orientation in the frame, distance from the starting manhole, defect severity score as well as number of defects in each category. Correspondingly, there are different deep learning tasks for sewer inspection with different level of complexity: (1) defect identification/classification of a whole image, (2) defect detection with bounding boxes to show type and location, (3) pixel-level defect segmentation to obtain quantitative evaluation. This Section briefly reviews related research from the above three aspects, in which deep learning methods are typically based on CNNs.

For defect classification, Kumar et al. proposed a framework by training a CNN model for classifying each type of defect respectively and achieved decent accuracy [6], which is not efficient for practical implementation since it requires multiple CNN models. Hassan et al. [45] classified six types of defects based on AlexNet architecture and applied optical character recognition (OCR) to generate the final inspection report. OCR is initially utilized by [28] to recognize text on the frame to extract defect information such as type and distance. Aside from normal deep learning, Xie et al. [46] proposed a two-level hierarchical deep learning method and achieved high classification accuracy. Meijer et al. [28] discussed the evaluation metrics for defect classification, and performed cross validation on their developed CNN model, which outperformed other SOTA model at that time.

As for defect detection, two-stage object detector such as faster region-based CNN (Faster R-CNN) is utilized to find the location of cracks, deposits and tree roots in CCTV images [8]

while one-stage method e.g. YOLOv3 is also applied [47]. Kumar et al. [6] compared three state-of-the-art models including Faster R-CNN, YOLOv3 and Single-Shot Multibox Detector (SSD) for multiple defect detection, among which faster R-CNN achieved the highest accuracy in their experiments. Li et al. developed an approach to address the potential influence from the data imbalance using CNN with hierarchical classification [48].

As pixel-level defect segmentation requires more efforts for data preparation and labelling, not many studies have been performed. Several researchers have applied CNNs for the segmentation of cracks REF, partly because more crack datasets are publicly available. In terms of sewer defect segmentation, Künzel et al. [49] proposed an approach for segmenting eight types of sewer defects in monocular fisheye images by revising the full-resolution residual network (FRRN). Meanwhile, Wang and Cheng [9] developed a unified network for segmenting cracks, deposits and roots in CCTV images by integrating CNN with dense CRF, which achieved good segmentation results.

It is noticed that the majority of existing research (either based on conventional computer vision or deep learning) has focused on classification, detection and segmentation of sewer defects or other components in a given image. Those tasks only recognize or segment defect on each image, but do not track defects or distinguish if the defect is the same one across consecutive frames. That means previous studies do not assign identity information to the detected defects, while assigning unique ID to defects could help count the number of defects in each category and assist with evaluating sewer condition. Therefore, an effective method to track defects in the video such as to provide ID for each defect needs to be investigated.

2.2. Multiple object tracking

Multiple object tracking (MOT) aims to identify and track objects across consecutive frames in the video. MOT has been an important research topic in the field of computer vision due to its potential applications in various areas, such as video surveillance, autonomous cars and crowd behavior analysis [50]. There have been different frameworks developed for visual tracking, such as discriminative correlation filters (DCF) [51,52], silhouette tracking [53,54], Kernel tracking [55,56] as well as point tracking [57]. However, the performances of the aforementioned methods are not satisfactory as expected due to the handcrafted feature representations e.g. histogram of oriented gradients (HOG), and Color-Names (CN) [58]. With the development of deep learning techniques and availability of computing power, various deep learning-based approaches are proposed for multiple object tracking. Most existing deep learning methods for MOT problems are typically designed based on the tracking-by-detection principle, by firstly using a detection model to detect objects in the image, and then employing an association model to extract object features (i.e. Re-ID features) and assign the detected objects to one of the existing tracks based on cost matrix [50]. The state-of-the-art MOT methods include simple online and realtime tracking (SORT) [59], multi-target tracking based on CNN features (CNNMTT) [60], simple online and realtime tracking with a deep association metric (Deep SORT) [61], most of which rely greatly on the detection accuracy and Re-ID feature quality.

As more visual technologies such as the surveillance camera and unmanned aerial vehicle (UAV) are employed in the construction industry, multiple object tracking has also been explored for monitoring construction site resources. Park et al. [62] classified vision-based

tracking methods into contour-based, point-based, and template-based methods, among which template-based methods were identified as most appropriate for construction entity tracking. Yang et al. [63] developed a tracking approach for tracking multiple workers on construction site by training a spatial model using machine learning. The developed approach incorporated functions to address occlusion problem, but required sample templates to initialize tracks. Konstantinou et al. [64] proposed a method for 2D tracking of workers exploiting color features to tackle several challenges such as abrupt movement, background clutter, and scene illuminations, which obtained decent tracking performance. The integration of detection and tracking was explored to improve tracking scenarios with occlusions. For example, Park and Brilakis [65] integrated detection and tracking for continuous localizing on-site workers, while Rezazadeh Azar et al. [66] combined detection and tracking by utilizing the HOG-based detection and the Kanade-Lucas-Tomasi (KLT) tracker, which performed well for tracking excavators and dump trucks even under partial occlusion. Similarly, detection and tracking are combined by using HOG and SVM for detection and particle filtering for tracking, leading to high recall and precision values for tracking workers and equipment [67]. In addition to 2D tracking, methods for 3D tracking are also studied. Brilakis et al. [68] developed a framework that performed 2D detection and tracking of construction entities in multiple cameras and generated 3D coordinates based on stereo vision. Lee and Park [69] presented a framework for 3D tracking of multiple onsite workers based on stereo vision and incorporated entity matching to find matches of tracked workers.

Most of the aforementioned methods for construction entity tracking are based on handcrafted features e.g. HOG and colors, along with typical classifiers such as SVM, the process of which still require massive manual efforts for designing feature descriptors and classifier parameters. In recent years, deep learning has attracted the attention for tracking construction site entities as well. Roberts and Golparvar-Fard [70] proposed a deep learning-based approach for detecting and tracking earthmoving equipment using CNNs. More recently, [71] developed a method for tracking multiple construction workers based on deep learning, which consists of detection, matching and re-matching. They claim that their deep learning approaches perform well in addressing tracking challenges, such as scale variations, appearance similarities, partial occlusions and posture variations. Nevertheless, the research of deep learning in tracking construction entities is still at its infancy and more methods needs to be investigated considering the availability of dataset and characteristics of construction site.

3. Defect tracking based on defect detection and metric learning

With the aim to track each defect and obtain the number of unique defects in the CCTV video, this paper proposes a framework for tracking multiple sewer defects. As shown in Figure 1, there are three main modules in the proposed framework, including (1) defect detection, (2) metric learning, and (3) defect tracking. The detection module is used to detect defects in each frame of the video using bounding boxes, and the results are used as inputs for the tracking module. Meanwhile, the metric learning module is aimed to train a model capable of extracting discriminative features of sewer defects for re-identifying defects in the video. In the end, the tracking module tracks defects by extracting features of detected defects through the metric learning model, and associating detections with tracks based on appearance features, motions as well as defect classes. For each module, there are similar procedures, including preparing

dataset, constructing the algorithm or model architecture, training, evaluating and saving the model, as well as implementing the model to generate desired results. Details of each module are introduced in the following.

Figure 1. The proposed methodology for tracking sewer defects based on defect detection and metric learning

3.1. Defect detection

Defect detection in the proposed framework is designed to detect the defects in each video frame and provide the inputs for the tracking process. As our method is based on the tracking-by-detection principle, the detection results could have great influence on the tracking performance. Considering the decent performance of deep learning for detecting sewer defects in our previous work [8], we develop a defect detection model based on a deep learning model named Faster R-CNN. The Faster R-CNN model originally developed by [72], was selected in this study because it was more accurate in detecting cracks, root intrusions, and lateral connections, compared to two other state-of-the-art models such as Single Shot Detector (SSD) and You Only Look Once (YOLO) [73]. The Faster R-CNN model is a two-stage detection model and the forward pass consists of three main steps, including (1) generating feature maps through the backbone CNN layers, (2) training a region proposal network (RPN) to produce regions of interest (ROIs) and propose candidate bounding boxes to indicate the presence of objects, (3) The regions in the image corresponding to i.e. for each ROI are processed through two fully connected convolutional layers to generate a class probability score and a bounding box adjustment factor, which attempts to better fit the bounding box to the object. Detailed architecture of the model can be referred to [8].

3.2. Metric learning

The objective of metric learning is to learn a metric which enables two extracted features of the same object to be closer than the features of different objects [74]. Metric learning is adopted in this study to learn discriminative features of different sewer defects, which can be used for re-identifying defects in the CCTV video. We propose a method based on omni-scale network (OSNet) [75] to learn the features of sewer defects using our dataset. OSNet is used because it is a state-of-the-art deep learning model for person Re-ID with decent performance

on public datasets. Particularly OSNet is a lightweight network and performs well on dataset containing objects with different scales, which is appropriate for sewer defect dataset with various sizes and scales. Details of the OSNet model can be referred to [75]. The original OSNet is designed for person Re-ID, and the model is trained on several state-of-the-art public dataset, such as Market 1501 [76], CUHK03 [77], and Duke [78]. To utilize the model for re-identifying sewer defects, we train the model using our sewer defect dataset, SewerTrack, which is extracted from CCTV videos and annotated with ground truth classes and ID information. Details of the dataset is introduced in Section 4.1. Considering that the choice of loss function will affect the training process and the final model performance [79], we train the model using different loss functions respectively and compare their performance. In addition, as our dataset size is relatively smaller than those for person Re-ID, we explore training the model using different data sources, which investigates the model genericity as well as the ability to transfer knowledge among datasets in different domains. Experiment settings and results are presented in Section 4.

3.3. Defect tracking

After training the detection model and the metric learning model, the two trained models are utilized to detect defects and extract discriminative features of each detected defect respectively. The proposed work process of the tracking module is illustrated in Figure 2. The tracking module works by (1) detecting defects in the video and extracting features for each detection, (2) initializing tracks with detected objects, predicting track motions in the following frame, (3) computing the distance metrics between detected defects and existing tracks as well as calculating cost matrix, and (4) associating detections with tracks based on cost matrix. Detailed methods are introduced in the following sections.

Figure 2. The proposed method of tracking multiple sewer defects

3.3.1. Defect detection and feature extraction

The proposed tracking method requires detected defects to initialize the track as well as to proceed the tracking process. By running the trained deep learning detection model on the video frames, defects on the image are detected with bounding boxes and the results for all the frames in the video are stored in a file, including the frame number, bounding box coordinates, detection confidence score, and the defect type. On the other hand, with the bounding box coordinates, the image patches containing the defects are cropped. Afterwards, the features of the cropped image patches are extracted using the metric learning model and stored in a file as well. Consequently, the detection results and the features of those defects are obtained and utilized in different part of the tracking module.

3.3.2. Track state prediction

Following the common practices for person tracking [61], the state of the track is defined using a set of parameters: $(x, y, r, h, \dot{x}, \dot{y}, \dot{r}, \dot{h})$, including the bounding box center coordinates (x, y) , the bounding box aspect ratio r , height h , and their velocities $(\dot{x}, \dot{y}, \dot{r}, \dot{h})$ in the image. Kalman filter is utilized to predict the track state based on observations of the object state (x, y, r, h) using the constant velocity motion and linear observation model. A track is initialized using detections that are not associated to any existing track. There are three possible track states including tentative, confirmed and deleted. New tracks are classified as tentative within their first three frames and are expected to be associated with certain detections during this time, otherwise, they will be deleted. For each tack, the number of frames since last successful association is counted as a_k , the value of which is increasing during the Kalman filter update process and is set to 0 when the track is successfully associated with a detection. Therefore, the track with a_k greater than the pre-defined threshold A_{max} is likely to have left the camera view and is hence deleted.

3.3.3. Distance metric calculation

To assign the detection with the track, the distance (indicating the similarity) between the detection and all the existing tracks is calculated, and the cost matrix of assigning the detections is obtained, based on which the association is performed using Hungarian algorithm (i.e. a common method for linear assignment problems). In addition to two types of distance metrics commonly used in object tracking, i.e. motion and appearance, we also integrate defect class information. The main reason is that the targets in most object tracking all belong to the same class while in the task of tracking defects, there are multiple classes of defects with different characteristics, which may provide clues for tracking. The calculation methods of the three distances are illustrated in the following.

Motion distance. The motion distance is calculated using the squared Mahalanobis distance because it measures the standard deviations between the detection and the mean track location, which can reduce the state estimation uncertainty in the tracking process. The distance between the detected objects and the state of existing tracks is calculated using Equation (1):

$$d^{(1)}(i, j) = (d_j - y_i)^T S_i^{-1} (d_j - y_i) \quad (1)$$

where d_j is the state of j^{th} bounding box, (y_j, S_i) is the projection of the i^{th} track distribution into the measurement space. After obtaining the Mahalanobis distance, a pre-defined threshold is utilized to filter the associations whose distance is larger than the threshold. According to the practices in pedestrian tracking, we set the threshold to be 9.4877 for the four dimensional measurement space.

Appearance distance. Mahalanobis distance is an appropriate association metric when the motion uncertainty is low, while the Kalman filter in the image-space only provides a rough estimation of object locations. Moreover, during sewer inspection, the camera motions heavily depend on the pipe condition, cameras may be suddenly zoomed or rotated once encountering defects. Considering such high uncertainties of the camera motions, only using Mahalanobis distance is not sufficient for tracking sewer defects. Hence, we take the appearance features of the defects into account when computing the distance between detections and tracks. Firstly, features of image patch contained in the j^{th} detected bounding box are extracted using the deep learning model trained in the metric learning part. The extracted feature vector r_j is regularized with $\|r_j\| = 1$. Meanwhile, to measure the appearance similarity between the detections and tracks, for each track k , the features of latest N detections ($N = 100$ in this study) that associated with the track are recorded in a set $R_k = \{r_k^{(i)}\}_{i=1}^{N_k}$. The appearance distance is measured by the smallest cosine distance between the j^{th} detection and the i^{th} track using:

$$d^{(2)}(i, j) = \min \{1 - r_j^T r_k^{(i)} \mid r_k^{(i)} \in R_i\} \quad (2)$$

Similar to the motion distance, another threshold is utilized to remove unqualified associations in terms of appearance distance.

Defect class distance. In addition to motion and appearance, this study also integrates the defect class information as an extra metric since different class of defects carries quite different features. There are three categories of objects for tracking in this study, including fractures, tree roots and laterals, which is encoded using integer $\{1, 2, 3\}$ respectively in the detection feature vectors. The class vector of the j^{th} detection is denoted as q_j . Similar to the appearance metric, the class vectors of the latest N detections associated with the track are kept in $Q_k = \{q_k^{(i)}\}_{i=1}^{N_k}$. The class distance is computed by the smallest absolute distance between the j^{th} detection and the i^{th} track:

$$d^{(3)}(i, j) = \min \{abs(q_j - q_k^{(i)}) \mid q_k^{(i)} \in Q_i\} \quad (3)$$

Cost matrix. In the end, the total cost matrix is constructed by integrating the distance metrics of motion, appearance and defect class. As shown in Equation (4) and (5), the total cost of assigning detection j to track i is computed by assigning a weight to each distance metric and summing them up.

$$c_{i,j} = w_m * d^{(1)}(i, j) + w_a * d^{(2)}(i, j) + w_c * d^{(3)}(i, j) \quad (4)$$

$$w_m + w_a + w_c = 1 \quad (5)$$

where w_m , w_a , w_c represents the weight for the metric of motion, appearance and class respectively. Our proposed integrated cost matrix aims to achieve decent tracking based on the useful information provided by three distance metrics. On the one hand, the Mahalanobis distance provides more clues about object locations based on motion, which is useful for short-term tracking. On the other hand, the cosine distance of the appearance can assist with long-term tracking by re-identifying objects that are occluded or missing for a while. Meanwhile, the defect class metric is expected to provide extra information to filter out unlikely associations. The weight assigned to each distance metric is used to control the influence of each metric on the tracking process. In this study, experiments are performed to investigate the optimal weights as well as the influence of different weight combinations on the tracking results.

3.3.4. Track association

Based on the distance metric and the cost matrix, Hungarian algorithm is applied to associate the detections with tracks, as it is a widely utilized method for assignment problems. Specifically, the inputs for the association include a set of tracks, detections and the maximum track age A_{max} . First, the cost matrix is calculated using methods introduced in Section 3.3.3. Then, we perform the association using linear assignment for all the tracks iteratively with an increasing track age, i.e. tracks with smaller age is associated first. Before linear assignment, we filter out unqualified associations whose distance metric value greater than thresholds. Afterwards, Hungarian algorithm is used to generate the preliminary association results, including the matched pairs, unmatched tracks and unmatched detections. The matched pairs are used to update the Kalman filter state to prepare for predicting next frame while the unmatched detections are used to initialize new tracks. In the end, the unmatched tracks with an age of 1 are associated again based on the intersection of union (IoU), which can prevent deleting tracks that are temporarily occluded. The unmatched tracks with age larger than 1 are deleted. Details of the association process are shown in Figure 3.

Figure 3. Flowchart of the proposed track association process

3.4. Evaluation Metrics

As the three modules in the proposed framework are designed with different purposes and mechanisms, we use different metrics to evaluate the performance of each module in the experiment with reference to practices in other research.

3.4.1. Evaluation of defect detection

In object detection, intersection over union (IoU) of the predicted bounding box and the ground truth box is commonly used to evaluate if an object is detected in an image. The detection is counted as true positive if the IoU is greater than a threshold (0.5 in this study) and the detection class is the same with the ground truth class. If one of the criteria is not met, then the detection is treated as false positive. Additionally, the ground truth boxes without required IoU with any detections are referred to as false negatives. Based on the above statistics, precision and recall can be calculated. In addition to IoU, the confidence score of each detection is used to threshold the detections as well. We use mean average precision (mAP) to evaluate the defect detection accuracy, which is one of the typical evaluation metrics for object detection. Details of the calculation method can be referred to our previous work [8].

3.4.2. Evaluation of metric learning

Performance of metric learning for image matching and Re-ID is usually evaluated using Rank-K accuracy and mAP. Given a query image in the image matching task, the algorithm retrieves a list of images ordered with similarity scores, Rank-K accuracy considers the result to be true positive if there is at least one image matched with the query image in the top K results. For example, Rank-1 accuracy means the algorithm only evaluates the first image in the ranked list to see if it has the same ID with the query image while rank-5 accuracy treats the image is correctly matched if there is at least one image among the first five retrieved images having the same ID with the query image. The final Rank-k accuracy is calculated as the ratio of the matched times over the total query times. We use Rank-1 accuracy which can better reveal the Re-ID ability. Following the common practices in person Re-ID [76], mAP is also included to evaluate the metric learning model. For each query, the area under the Precision-Recall curve is calculated as the average precision (AP) and then the mAP is obtained by computing the mean of the AP values across all the queries. The method to calculating mAP is similar to that is used for the detection model.

3.4.3. Evaluation of defect tracking

In our tracking experiments, after mapping the detections with tracks in each frame using the methods introduced in Section 3.3.4, the ground truth detections that are not associated with a track are counted as false negatives (FN), and the generated tracks that are not associated with any ground truth detection are recognized as false positives (FP). In addition, the case that the ID of a tracked ground truth object is incorrectly changed during the tracking process is counted as an ID switch (IDs), also called mismatches. Typically, CLEAR MOT metrics developed for the Classification of Events, Activities and Relationships (CLEAR) workshops are commonly utilized for evaluating tracking algorithms, in which the Multiple Object Tracking Accuracy (MOTA) is one of the main metrics [80]:

$$MOTA = 1 - \frac{\sum_t(FN_t + FP_t + mme_t)}{\sum_t(FN_t + TP_t)} \quad (5)$$

where FN_t , FP_t , TP_t , mme_t are the number of false negatives, false positives, true positives and mismatches at frame t respectively. However, MOTA belongs to event-based measures which considers the number of ID switches and can assist with finding the source of mistakes, yet in some cases such as sports, security and surveillance, it is more desired to have a tracker that is capable of preserving the identify and returning the correct ID in the end [50,78]. Particularly, the aim of tracking defects in our study is to find the unique defect number in the video where obtaining the defect ID is more important, making the MOTA accuracy not suitable to measure the tracking performance in this paper. Alternatively, we use the ID scores [78], which are more appropriate for evaluating how well the predicted identifies conform to true identities. The ID scores are calculated based on ID precision and ID recall and IDF1 is mostly used for eventual performance evaluation:

$$IDP = \frac{\sum_t TP_t}{\sum_t (FP_t + TP_t)} \quad (6)$$

$$IDR = \frac{\sum_t TP_t}{\sum_t (FN_t + TP_t)} \quad (7)$$

$$\text{IDF1} = \frac{2}{\frac{1}{\text{IDP}} + \frac{1}{\text{IDR}}} = \frac{2 \sum_t TP_t}{\sum_t (2TP_t + FP_t + FN_t)} \quad (8)$$

In our experiment, we use IDF1 score as the main metric to evaluate the tracking performance for counting the unique defect number. Meanwhile, we also incorporate the ID switch number to assess the ability of the tracker to track the defects in different positions.

4. Experiments and Results

In this section, we perform experiments to validate the whole framework for tracking sewer defects in videos, as well as to explore the influence of configurations of the three modules on the tracking performance.

4.1. Dataset preparation

To evaluate the whole framework as well as each individual module, we collect and prepare different datasets with the required formats and annotations.

4.1.1. Dataset for the defect detection

The Faster RCNN object detection model was trained and validated using 3,600 images (i.e. 1,200 images of fractures, root intrusions, and lateral connections respectively) in which the observations were manually identified using rectangular. The images originate from 8-inch, 10-inch, and 12-inch diameter vitrified clay pipes located in Florida and Georgia. The images had been recorded using front facing CCTV cameras and had resolutions between 1507×720 and 720×576. An open source image annotation tool called LabelImg, was used to facilitate the drawing of bounding boxes on images [81]. **Figure 4 illustrates the examples of the annotated defect images.**

Figure 4. Example of the image annotation for training the defect detection model

The images were labeled by two Civil Engineering undergraduate students. To improve consistency in the labeled images approximately 50 random images were initially selected and presented to both individuals. To ensure inter-coder reliability, the annotations were compared to highlight potential inconsistencies in the labeling of images. 90% of the data set (i.e. 3,240 images) was used for training and 10% (i.e. 360) was used for evaluating the models. The distribution of training and validation images are shown in Table 1. The training and evaluation sets of images were selected using a randomized algorithm to reduce biases in the data set. Training and validation were performed on a desktop computer comprising of an Intel CPU and an NVIDIA P4000 Quadro CUDA supported graphics card with 8GB memory. The Faster RCNN model took approximately 12 hours to train.

Table 1. Number of images in training and validation for the detection model

Image Category	Training	Validation	Total
Fractures	1080	120	1200
Root Intrusions	1080	120	1200
Lateral Connections	1080	120	1200
Total	3240	360	3600

4.1.2. Dataset for metric learning

To train the metric learning model for re-identifying sewer defects in CCTV videos, we extracted images from videos which contain different types of defects. When encountering a defect, the cameras on the CCTV inspection crawler usually zoom the camera and rotate the camera up and down around the defect area to inspect the defect. Such operation of the camera typically leads to a series of frames containing the same defect, yet with different features, such as the scale, orientation, illumination and colors. Those consecutive frames with the same defect are extracted and categorized as the images with the same defect ID. Using a tool called CVAT [82], each unique defect in the video is annotated with a bounding box, the defect type and an ID number. After annotation in CVAT, we obtained the XML files containing the ground truth information of the defects in those videos.

To generate the dataset for training the metric learning model, we crop the image patches inside the bounding boxes in each frame using the box coordinates, and extract other information from the XML files. Then we rename the cropped images according to the naming rules of the Market1501 [76], which is a state-of-the-art (SOTA) dataset for person Re-ID. For metric learning, three types of datasets are required, i.e. training dataset, testing (also called gallery) dataset and query dataset. After model training, testing is performed by extracting images from the testing set to match the images in the query set. In our dataset, there are 107 unique defects with 10,367 images in total and we split the dataset into train and test with a ratio of 7:3 in classes, i.e. 75 classes in train set and 32 classes in test set. For the images of each class in test set, we extract two images per class as the query set. Consequently, there is no overlap of classes between train and test while the classes in test and query are the same (with different images). Market 1501 [76] and CUHK02 [77] are two SOTA public datasets for person Re-ID, which are also used in our experiments in Section 4.3.2 to investigate the influence of different datasets on metric learning and tracking results. Details of our dataset SewerTrack and the other two SOTA datasets are listed in Table 2.

Table 2. Details of our dataset and other SOTA dataset for Re-ID

Name	Dataset	No. of IDs	No. of Images	No. of Cameras
SewerTrack (Ours)	Train	75	6,135	2
	Query	32	64	2
	Test	32	4,168	2
	Total	107	10,367	2

Market1501 [76]	Train	751	12,936	6
	Query	750	3,368	6
	Test	751	15,913	6
	Total	1501	32,217	6
CUHK02 [77]	Train	1577	6,308	8
	Query	239	478	1
	Test	239	478	1
	Total	1816	7,264	8

4.1.3. Dataset for defect tracking

To validate the proposed method for tracking sewer defects in videos, we prepare the dataset according to the format of MOT benchmark datasets [76]. For each video to track, the tracking algorithm requires (1) detection files generated from the defect detection model mentioned in Section 3.1, (2) features of each defect image patch extracted by the trained metric learning model mentioned in Section 3.2, (3) frame images of the video, and (4) ground truth files to evaluate the tracking performance. First of all, the detection results are obtained and saved to files by running the trained detection model on each video. In terms of the ground truth file for tracking, similar to the annotation for the metric learning dataset, we annotate the inspection videos using CVAT to generate the XML files containing defect information, after which the required ground truth information is retrieved for evaluating tracking. One example of the ground truth file and detection result file is illustrated in Figure 5. Both files include the frame number, bounding box data (x,y,w,h) where (x,y) is the top left point coordinate and (w,h) is the box width and height, as well as the defect class index. The main difference is that the ground truth file for tracking includes the unique defect ID while the detection file incorporates the detection probability.

	Frame Number	Defect ID	Bbox (x,y,w,h)	Defect class index	Detection probability
1	1	1	387.900,31.290,44.480,143.160	1	-1,-1,-1
2	1	2	418.500,17.500,177.480,208.680	1	-1,-1,-1
3	2	1	387.850,31.300,44.480,143.160	1	-1,-1,-1
4	2	2	418.420,17.520,177.480,208.690	1	-1,-1,-1
5	3	1	387.800,31.310,44.480,143.160	1	-1,-1,-1
6	3	2	418.340,17.550,177.480,208.680	1	-1,-1,-1
7	4	1	387.750,31.320,44.480,143.160	1	-1,-1,-1
8	4	2	418.250,17.570,177.490,208.680	1	-1,-1,-1
9	5	1	387.700,31.330,44.480,143.150	1	-1,-1,-1
10	5	2	418.170,17.600,177.490,208.680	1	-1,-1,-1
11	6	1	387.650,31.330,44.480,143.160	1	-1,-1,-1
12	6	2	418.090,17.620,177.480,208.680	1	-1,-1,-1
13	7	1	387.600,31.340,44.480,143.160	1	-1,-1,-1
14	7	2	418.010,17.650,177.480,208.680	1	-1,-1,-1
15	8	1	387.550,31.350,44.480,143.160	1	-1,-1,-1
16	8	2	417.920,17.670,177.490,208.680	1	-1,-1,-1

(a) Ground truth file

	Frame Number	Defect ID	Bbox (x,y,w,h)	Defect class index	Detection probability
1	1	1	-1,376.633,18.366,71.264	173.650,0.990103	1
2	2	1	-1,378.048,17.724,68.489	170.931,0.974647	1
3	3	1	-1,377.579,13.381,68.649	179.155,0.952554	1
4	3	1	-1,24.339,0.000,129.844	154.304,0.762505	1
5	4	1	-1,376.209,12.471,70.140	175.752,0.975261	1
6	5	1	-1,377.625,16.969,68.279	170.563,0.979809	1
7	6	1	-1,377.335,16.517,69.309	169.317,0.991239	1
8	6	1	-1,27.622,0.023,126.417	153.113,0.742872	1
9	7	1	-1,379.152,17.280,67.899	173.305,0.976500	1
10	7	1	-1,27.804,1.294,125.953	152.204,0.745349	1
11	8	1	-1,378.390,17.597,67.744	165.756,0.992715	1
12	9	1	-1,378.400,13.694,67.185	175.891,0.990169	1
13	10	1	-1,379.776,20.377,67.338	162.201,0.990318	1
14	11	1	-1,380.976,19.813,66.017	162.208,0.991454	1
15	12	1	-1,380.109,19.010,67.117	169.050,0.990185	1
16	13	1	-1,380.768,20.159,68.460	168.821,0.969239	1

(b) Detection result file

Figure 5. Example of the ground truth file and detection result file used in the tracking dataset

4.2. Influence of defect detection

The deep learning model based on Faster R-CNN for sewer defect detection is constructed and trained using the dataset introduced in Section 4.1.1. As a result, the trained model achieves an average precision (AP) of 0.62, 0.73 and 0.95 for detecting fractures, root intrusions and lateral connections respectively, leading to an mAP of 0.77 in the end. The trained model with

the highest accuracy is implemented to generate defect detections for the videos, and the detection results are used in the tracking process.

To evaluate the influence of the detection accuracy on the tracking performance, we first generate two sets of detections for the 23 tracking videos, including the detection results by running the trained Faster R-CNN model as well as the ground truth detections. Afterwards, these two sets of detections are used as the inputs to implement our proposed tracking algorithm respectively. The tracking performance is evaluated in terms of the average IDF1 score and the ID switch number of all the videos. As shown in Table 3, the average IDF1 of testing videos obtained based on the ground truth detections is 94.4%, which is much higher than using detections from the Faster R-CNN model. Meanwhile, tracking using the ground truth detections achieves much fewer ID switches compared with that based on the detections generated by the detection model. The experiment demonstrates our proposed framework is capable of tracking sewer defects in videos with a state-of-the-art IDF1. On the other hand, the comparison reveals the significant influence of the detection accuracy on the tracking results and reminds us the importance of improving detection model to ultimately facilitate tracking process.

Table 3. Tracking accuracy using detections from detection model and ground truth

Detection source	IDF1 (%)	IDs
Defect detection model	56.4	203
Ground truth detection	94.4	7

4.3. Influence of metric learning

The aim of this experiment is to verify the effectiveness of metric learning model and to investigate the influence of the training settings of the metric learning model on the Re-ID performance and tracking results. Specifically, the influence of different loss functions as well as the training dataset are studied in the following.

4.3.1. Metric learning with different loss functions

The metric learning model introduced in Section 3.2 is trained with our annotated sewer dataset for re-identifying defects in videos and extracting discriminative features of sewer defects that are used in tracking process. As the loss function has great influence on the training process as well as the final accuracy of the model [79], we train our model using three loss functions respectively, including the cosine softmax, triplet loss and magnet loss, as they have different working mechanisms and are commonly utilized for person Re-ID.

Cosine softmax loss. The standard softmax classifier is dominantly utilized for classification tasks by placing a softmax classifier on the last layer of the network to generate the probability score for each class using Equation (9).

$$p(y = k | s) = \frac{\exp(w_k^T s + b_k)}{\sum_{n=1}^C \exp(w_n^T s + b_n)} \quad (9)$$

where $s = f(x)$ is the feature representation from an encoder network, w is the weight of the network, b is the bias, and y is the class label. In this study, we use the cosine softmax classifier which can generate more compact clusters by modifying the standard softmax classifier according to [74]. The modifications include applying L2 normalization to the final layer of the encoder network, i.e. $\|f(x)\|_2 = 1, \forall x \in \mathbb{R}^D$, as well as normalizing the weights to unit-length, i.e. $\tilde{w}_k = w_k / \|w_k\|_2, \forall k = 1, \dots, C$. Consequently, the cosine softmax is defined as:

$$p(y = k | s) = \frac{\exp(\alpha \cdot \tilde{w}_k^T s)}{\sum_{n=1}^C \exp(\alpha \cdot \tilde{w}_k^T s)} \quad (10)$$

where α is a free scaling parameter. Compared with the standard softmax, the cosine softmax has fewer parameters as the bias terms are removed. Eventually, the parameters are obtained by minimizing the cross entropy loss through Equation (11).

$$L = \begin{cases} -\sum_{i=1}^N \sum_{k=1}^C \log p(y_i = k | s_i), & y = k \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

Triplet loss. Triplet loss [83] is developed based on tuples of (s_a, s_p, s_n) where s_a is the representation of the baseline (anchor) input, s_p is the positive input representation with the same class as s_a , s_n is a negative input representation with different class from s_a . For each tuple, the loss function aims to make the positive pair $\{s_a, s_p\}$ close to each other while negative pair $\{s_a, s_n\}$ far away from each other. Hence, the loss requires the distance between positive pair and negative pair larger than a margin m :

$$L(s_a, s_p, s_n) = \{\|s_a - s_n\|_2 - \|s_a - s_p\|_2 + m\}_+ \quad (12)$$

$$g(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (13)$$

where $\{\}_+$ represents the hinge function shown in Equation (13), which equals to 0 if input is negative and identity otherwise.

Magnet loss. Instead of working on individuals, pairs or tuples of samples, magnet loss [84] focuses on entire regions of the class distribution. The loss function models the distribution of different classes and aims to reduce the overlap between distributions by separating each sample from the means of other classes. The original magnet is designed for multi-modal metric learning, and a simplified version is applied in this study for the sewer Re-ID task:

$$L(y, s) = \left\{ -\log \frac{e^{-\frac{1}{2\sigma^2} \|s - \hat{\mu}_y\|_2^2 - m}}{\sum_{k \in \bar{C}(y)} e^{-\frac{1}{2\sigma^2} \|s - \hat{\mu}_k\|_2^2}} \right\}_+ \quad (14)$$

where $\bar{C}(y) = \{1, \dots, C\}$ excluded $\{y\}$ is the class set in the sample, $\hat{\mu}_y$ is the mean value of samples of class y , $\hat{\sigma}^2$ is the variance of all the samples that are away from their class mean, s is the sample representation, m is the margin.

During the training process, the loss curve of each model with different loss function is plotted, as shown in Figure 6. It can be seen that the model with the cosine softmax function drops quickly within 20,000 iterations and converges with a plateau value of 0.6 after that. Similarly, the loss of the model with magnet loss function decreases with a fast rate and almost reaches zero within around 20,000 iterations. Although there are some small fluctuations after that, the loss curve remains at zero, which is much smaller than that of the model with cosine softmax. On the contrary, the loss of the model trained with triplet loss decreases gradually at a slower speed in the whole process, with a loss value of 0.24 in the end. The comparison demonstrates that training of the model with cosine softmax is more stable and converges fast while the model using magnet loss also converges quickly, with a smaller loss value. On the other hand, the model with the triplet loss may be harder and slower to achieve convergence.

Figure 6. Loss curves of metric learning model with different loss functions

To evaluate the influence of the metric learning model on the tracking performance, each of the model trained with different loss function is applied to the tracking process by generating features of the detected defects, which are used to calculate the cost matrix and hence affect the results in tracking. As shown in Table 4, tracking using the features generated by the model trained with magnet loss achieves the highest IDF1 of 57.4% while using features from the model trained with cosine softmax is the lowest. Tracking based on the model trained with triplet obtains similar performance with slight lower accuracy than that with magnet. Correspondingly, the ID switch number drops when using the model trained with triplet and magnet. The results indicate that models trained with triplet or magnet are more likely to find discriminative features for re-identifying defects in videos, such as to improve IDF1 and reduce the ID switches.

Table 4. Tracking accuracy using metric learning model trained with different loss functions

Loss function	IDF1(%)	IDs
cosine	56.40	203
triplet	57.20	196
magnet	57.40	202

4.3.2. Metric learning using datasets from multiple domains

Considering the importance of dataset characteristics and size for most deep learning models, this experiment aims to explore the possibility of using same-domain and cross-domain datasets, i.e. dataset from different domains, to enrich the training dataset for the metric learning model. We evaluate the influence of different datasets on the metric learning model as well as the inherent influence on tracking results. To this end, we train four metric learning models using the individual datasets introduced in Section 4.1.2 and their combinations, including (1) Market1501 (2) our sewer dataset - SewerTrack, (2) combination of SewerTrack and Market 1501, (3) combination of SewerTrack, Market1501 and CUHK02. After training, all the models are tested on the testing dataset of SewerTrack.

The performance of the metric learning model is evaluated using the cumulative matching characteristics (CMC) Rank-1 accuracy and mAP. During the image matching process, the distance is calculated based on the cosine distance or Euclidean distance between the feature vectors extracted from the last FC layer of the query image and the testing image. As shown in Table 5, the testing accuracy of models trained with four different datasets is evaluated based on both Euclidean distance and cosine distance. The results show that the model trained merely with Market1501 achieves lower accuracy (in both mAP and Rank-1) than the model trained with the SewerTrack data, which is reasonable as the data from the same domain (sewer defect) carry similar characteristics with the query images, making the model easier to extract images with the same class from the gallery. Meanwhile, the model trained with the combination data of SewerTrack and Market1501 greatly outperforms the first two models with 6% improvement in mAP and Rank-1 accuracy. The results indicate that fusing data of other domains (i.e. pedestrian data) and the target domain (i.e. sewer data) to train the model can significantly boost the model performance. The possible reason is that such combination greatly enriches the training data, allowing the model to learn a variety of low-level features. On the other hand, adding the data of the target domain makes the model familiar with the characteristics of the target data, hence contributing to the matching accuracy. However, it is noticed that training the model using the combination of three datasets by adding extra CUHK02 data did not further increase the model accuracy compared with combining SewerTrack and Market1501, although there is still improvement in comparison with only using individual dataset (Market1501 or SewerTrack). Such results reflect that integrating datasets from other domain and the target domain can improve the model training and accuracy to a certain degree, but the improvement may not be proportional to the dataset size. Instead, training based on data from multiple domains makes the model exposed to a broad range of features, probably leading to more confusion and errors for matching images. In addition, we also notice that although the accuracy values based on cosine distance are in most cases higher than that using the Euclidean distance, evaluation results based on both distance metrics have similar observations as introduced above.

Table 5. Accuracy of the metric learning model trained with different datasets

Dataset	Euclidean Distance		Cosine Distance	
	mAP (%)	Rank-1 (%)	mAP (%)	Rank-1 (%)
Market1501	50.50	73.40	50.70	71.90
SewerTrack	61.20	78.10	62.30	81.20

SewerTrack+Market1501	67.10	85.90	68.00	84.40
SewerTrack+Market1501+CUHK02	62.40	78.10	64.30	78.10

In addition to evaluating the accuracy for image matching, this experiment also investigates the influence of the model trained with four different datasets on the tracking results. Similar to Section 4.3.1, after training the models, we apply the trained model to generate four sets of features of the detected defects respectively, and the generated features are then utilized in the tracking process. Table 6 shows the tracking results. It can be seen that the differences of using different models for tracking defects are not as obvious as that for matching images. Nevertheless, using the model trained with the target domain data, SewerTrack, still achieves slightly higher tracking accuracy compared with only using the Market1501 data. Meanwhile, it is interesting to find that tracking based on the model trained with three datasets outperforms other models, especially in terms of the ID switch number. Although such result is not totally consistent with the model performance for image matching, it still reveals the potential of fusing data from multiple domains to extract discriminative features to improve the Re-ID ability and reduce the confusion between different classes during the tracking process. This experiment also reflects that the metric learning model performing best in image matching may not always obtain the highest accuracy in tracking. The likely reason is that the image matching process is relatively static while the tracking process is more dynamic and complex, e.g. the extracted features can affect re-identifying objects, which will further influence updating the tracker and the final results.

Table 6. Tracking accuracy based on features generated from metric learning models trained with different datasets

Dataset	Tracking Accuracy	
	IDF1 (%)	IDs
Market1501	56.20	101
SewerTrack	56.40	95
SewerTrack+Market1501	56.10	97
SewerTrack+Market1501+CUHK02	56.40	92

4.4. Influence of the weight to the distance metric

This experiment aims to study the influence of the parameters used in the tracking algorithm. As mentioned in Section 3.3, different from most existing tracking methods, our proposed tracking algorithm is based on the cost matrix by summing up three weighted distance metrics, including motion, appearance, as well as the defect class, the last of which is rarely considered in other research. The weight assigned to the distance metric reflects the importance of each aspect in the tracking process and consequently affect the tracking accuracy. Therefore, the weight of each distance needs to be configured carefully. This experiment explores the impact and sensitivity of the three distance aspects on the tracking performance using different weights $\{w_m, w_a, w_c\}$ for motion, appearance and defect class respectively. All the following testing experiments are performed based on the same set of detections as well as features from the same metric learning model.

Influence of w_m . When investigating the influence of the motion distance for the tracking process, we fix the value of w_a and w_c respectively (i.e. $w_a = 0$, $w_c = 0$ in this study) and assign w_m with a value between 0 and 1 with an increasing interval of 0.1. Then we run the tracking algorithm with each set of weights and evaluate the accuracy using IDF1 and ID switches (IDs). As shown in Table 7, when w_a is fixed, the tracking algorithm achieves the highest IDF1 of 53.2% with $w_m = 0.2$ and the smallest IDs of 110 when $w_m = 0.1$. Similarly, in the case of fixing the value of w_c , the best performance is obtained with both the highest IDF1 and the smallest IDs when $w_m = 0.1$. Figure 7 shows the trend of IDF1 and IDs with the increasing of w_m . It can be seen that the IDF1 score is decreasing consistently with the rise of w_m in both cases of fixed w_a and w_c . Meanwhile, as shown in Figure 8 (b), the IDs number is generally increasing when w_m is getting larger, which means the Re-ID accuracy is dropping in the process. Overall, the results indicate that using larger weight for the motion distance may lead to poor tracking performance in terms of both IDF1 and IDs.

Table 7. Tracking accuracy with an increasing w_m value

w_m	w_a	w_c	IDF1 (%)	IDs	w_a	w_c	IDF1 (%)	IDs
0.1		0.9	52.80	110	0.9		53.20	104
0.2		0.8	53.20	128	0.8		51.20	117
0.3		0.7	51.30	144	0.7		51.30	136
0.4		0.6	51.30	158	0.6		50.90	148
0.5	0	0.5	50.60	182	0.5	0	50.60	159
0.6		0.4	49.90	176	0.4		49.80	177
0.7		0.3	49.80	175	0.3		50.00	178
0.8		0.2	49.00	181	0.2		48.80	179
0.9		0.1	49.10	181	0.1		49.10	182

Figure 7. Tracking accuracy with the increasing of w_m (a) IDF1 score (b) ID switch (IDs) number

Influence of w_a . Similar to the settings of studying the influence of w_m , to understand the impact of w_a , we assign an increasing value to w_a while keep the value of w_m and w_c unchanged respectively in two groups of experiments. The tracking accuracy of the two groups based on different combinations of weights is listed in Table 8. Different from the two

comparison groups that obtain similar results for evaluating the influence of w_m , there are more deviations between the tracking accuracies in the case of fixing w_m or w_c . In the case of using fixed w_m , the highest accuracy is achieved with an IDF1 of 57% when $w_a = 0.1$, which is much higher than the best IDF1 score (i.e. 53.2%) in the other group with $w_a = 0.9$. Similar phenomenon is also observed in terms of ID switches, with the best ID switch number of 88 and 104 respectively in two groups. It is noticed that even the worst performance in the first group has better accuracy than the best results in the second group, indicating the huge gaps between the two settings. Correspondingly, from the variation trend of the accuracy with the increase of w_a in Figure 8, we can see that the IDF1 curve in the case of $w_m = 0$ is always higher than that in the case with $w_c = 0$ while the IDs curve is consistently lower. Such results indicate that lower w_m value may assist with improving tracking performance. In addition to the gaps between the accuracy in two groups, we also find that when $w_c = 0$, IDF1 is growing consistently with the increase of w_a and IDs is decreasing gradually as well. On the other hand, in the case of $w_m = 0$, the curves of both IDF1 and IDs remain more stable with slight fluctuations in values. Such results reflect that the tracking accuracy is more sensitive to the object motion variation, and larger motion distance weight may degrade the tracking. Inversely, larger weight assigned to the appearance distance is likely to facilitate the tracking process.

Table 8. Tracking accuracy with an increasing w_a

w_a	w_m	w_c	IDF1 (%)	IDs	w_m	w_c	IDF1 (%)	IDs
0.1		0.9	57.00	97	0.9		49.10	182
0.2		0.8	57.00	97	0.8		48.80	179
0.3		0.7	56.70	90	0.7		50.00	178
0.4		0.6	56.90	88	0.6		49.80	177
0.5	0	0.5	56.80	90	0.5	0	50.60	159
0.6		0.4	56.80	95	0.4		50.90	148
0.7		0.3	56.70	93	0.3		51.30	136
0.8		0.2	56.60	94	0.2		51.20	117
0.9		0.1	56.30	98	0.1		53.20	104

Figure 8. Tracking accuracy with the increasing of w_a (a) IDF1 score (b) ID switch (IDs) number

Influence of w_c . The experiment settings for studying the influence of w_c are similar to that for the other two weights mentioned above. With an increasing value of w_c , we perform the

tracking algorithm in two groups with fixed w_a and w_m respectively. According to the tracking accuracy in Table 9, when w_c equals to 0.8, the highest IDF1 score is achieved in both groups with a value of 53.2% and 57% respectively. Meanwhile, the best IDs number is 110 when $w_c = 0.9$ in the first group and 88 when $w_c = 0.4$ in another group. Similar to the gaps observed in Figure 9, there are obvious accuracy deviations between the cases with fixed w_a and w_m , as illustrated in Figure 9. In the case of $w_m = 0$, the tracking performance is all the time better with higher IDF1 and smaller IDs than that in the case of $w_a = 0$. It is also observed that with the increase of w_c , the accuracy in the case of $w_a = 0$ is rising correspondingly with increasing IDF1 and decreasing IDs. At the same time, with $w_m = 0$, although the IDF1 curve is generally increasing, the increasement is extremely small, which is similar to the observation in the above experiment of investigating the influence of w_a . Similar conclusion can be drawn that tracking performance is more sensitive to motion variations. In addition, a larger weight to the class distance metric is likely to improve the tracking accuracy.

Table 9. Tracking accuracy with an increasing w_c

w_c	w_a	w_m	IDF1 (%)	IDs	w_a	w_m	IDF1 (%)	IDs
0.1		0.9	49.10	181	0.9		56.30	98
0.2		0.8	49.00	181	0.8		56.60	94
0.3		0.7	49.80	175	0.7		56.70	93
0.4		0.6	49.90	176	0.6		56.80	95
0.5	0	0.5	50.60	182	0.5	0	56.80	90
0.6		0.4	51.30	158	0.4		56.90	88
0.7		0.3	51.30	144	0.3		56.70	90
0.8		0.2	53.20	128	0.2		57.00	97
0.9		0.1	52.80	110	0.1		57.00	97

Figure 9. Tracking accuracy with the increasing of w_c (a) IDF1 score, (b) ID switch (IDs) number

4.5. Model performance

4.5.1. Model inference time

Conceptually, there are two methods to count the number of defects in an inspection video. The first one is to detect the defects in each frame and then manually identify if the defect is the same one across multiple frames. The second one is tracking the defect in each frame directly, which is proposed in this study. We compare the two methods in terms of the time

required for counting the defects, as shown in Table 10. The time is estimated to be at least 1000ms per image for manually identifying if the defect is the same one in multiple frames. The time for detection, feature extraction and tracking is the average time for each frame in all the testing videos during the experiments. As a result, the time for our method is 128.79 ms/image while it is 1110 ms/image for the method based on defect detection and manual identification. Although the defect tracking method is more complex than only performing the defect detection, the time required for the two methods still reveals the advantage of defect tracking.

Table 10. Time (ms per image) required by two different defect counting methods

	Counting through defect detection		Counting through defect tracking	
Required time (ms/image)	Detection	110	Detection	110
	Manual identification	1000	Feature extraction	16.86
			Tracking	1.93
Total time (ms/image)		1110		128.79

4.5.2. Model performance under different camera motions

During inspection, there can be several camera motions to better evaluate the defects, such as moving forward, panning to pipe wall, moving along the pipe wall and rotating around the defect. Some of the tracking results under different camera motions are extracted from the testing videos. In each video frame, the defects are tracked using the bounding box in different colors with the defect type and ID. Figure 10 shows the video frames where the camera is moving forward and crossing more than pipe segments. In Figure 10 (a), the tracker keeps a tight bounding box on the defects when moving forward, and remains the tracking status across two pipe sections until the defect disappears. Meanwhile, in Figure 10 (b) and (c), there are more defects detected and tracked in the process where the trackers perform well in keeping consistent ID number for existing defects and tracking new defects at the same time.

On the other hand, when the camera pans to the left pipe wall after discovering the tree root in Figure 11 (a), there are several blurry moments during the camera motion, but the tracker can still track the defect closely in the process. Similarly, Figure 11 (b) shows the camera panning along the pipe wall to further inspect the tree root, and our model successfully tracks the defect from when only part of the defect is included till the whole defect appears in the camera. In Figure 11 (c), the camera motion is more complex as it moves to another spot after detecting the fracture and then returns back to the defect. Under such circumstance, the tracker is still capable of re-identifying the defect with the same ID, which indicates the effectiveness of our proposed method for tracking and re-matching defects based on appearance features.

When dealing with the situation of camera rotating, our method also performs quite well. As shown in Figure 12, the circumferential fracture is distributing around the pipe wall, so the camera rotates up and down to assess the defect. Although the defects may appear different during the camera rotation, the tracker still successfully track them most of the times without losing their identities. Overall, the qualitative results demonstrate that our proposed framework is able to achieve a good performance for tracking multiple sewer defects even with complex camera motions.

Figure 10. Example tracking results under the "move forward" camera motion

Figure 11. Example tracking results under the "pan over" camera motion

Figure 12. Example tracking results under the "rotate up and down" camera motion

4.5.3. Model performance under different conditions

As the space inside the sewer pipes is dark by nature, lighting device is usually mounted on the inspection equipment to help inspectors see the pipe interior more clearly. The continuous movement of the equipment commonly leads to uneven illumination as well as blurry moments in the captured video, which may have influence on the tracking process. This section lists several examples of the tracking results under different conditions, including bright, dark and blurry videos.

When the camera is moved closer or zoomed to a defect, the lighting around the defect will be very bright. Figure 13 (a) and (b) shows the model is still be able to continuously track the tree roots around the pipe joint as well as the fractures despite of the very bright illumination. It is found that in most cases, the tracking model performs well when encountering bright conditions. However, it is also observed that the defect is ignored or misclassified due to the lighting condition. For example, the tree root at right-top of the pipe joint in Figure 13 (c) is identified as fracture or even ignored. The possible reason is that the strong light tends to erase the main part of the tree root while makes the edge more obvious, which is similar to the shape of a fracture.

Figure 13. Tracking under very bright condition

Another case is that when the defect is captured in the dark area, it is difficult for the model to discover the defect or correctly identify the defect. Figure 15 presents that the lateral connection is ignored in the first image as the background is too dark and it is partly occluded by the tree root. However, after it becomes brighter, the lateral is successfully detected in the second image. Figure 14 (c) and (d) shows that the fracture in the dark corner is identified as

tree root, which is actually not easy for human beings to discover it as well. It can be seen that dark condition tends to have more negative influence on the tracking performance compared with very bright condition. The main reason can be that weak illumination makes it more difficult for the model to extract useful features to discover and differentiate the defects.

Figure 14. Tracking under dark condition: (a)(b) lateral connection ignored, (c)(d) fracture misclassified as root

In the process of rotating or moving the camera, the video frames may become blurry. Figure 15 (a) demonstrates that the model is still capable of tracking the tree root continuously even when the video is quite blurry as the camera is panning across the pipe wall. Similarly, Figure 15 (b) shows that, in another video, despite of several blurry moments, our method keeps a close tracking of the tree root. However, it is also noticed that the model may fail to identify defects correctly when the video becomes extremely blurry. As shown in Figure 15 (c), the model detects a fracture incorrectly as a tree root due to the high degree of blurry. The fast movement of the camera as well as the exposure to strong lights make it difficult to recognize the fracture but turns the texture similar to that of a tree root, which makes the model confused. It is found that the model is able to track the defect in most blurry cases, especially when the defect has relatively large area, such as the tree root or deposit. Meanwhile, for defects with tiny or thin shapes (e.g. fractures) or when the blurry is serious, the model is more likely to ignore the defect or misclassify them into another type.

In summary, the tracking model generally performs well (i.e. tracking the defect continuously) in very bright or blurry situation. Nevertheless, extreme blurry moments also affect the detection accuracy and thereby makes the model fail to track the defects. On the other hand, the model is more susceptible to dark environment, in which extracting defect features is difficult. Therefore, methods need to be investigated to reduce the influence of illumination and blurry condition on detection and tracking process.

(a) Successful tracking in blurry video

(b) Successful tracking in blurry video

(c) Misclassification in blurry video

Figure 15. Tracking in blurry videos

4.6. Discussion

Overall, the experiment results demonstrate the capability of our proposed framework to track multiple sewer defects in inspection videos with a decent IDF1 score. The qualitative results show that our model is able to track defects in most videos even under different camera motions and environment conditions. In addition, the quantitative results in the experiments indicate the performance of each module of the framework can affect the overall tracking accuracy to a certain degree. First of all, we adopt the tracking-by-detection method in our framework, making the tracking accuracy greatly rely on the detection accuracy. Tracking based on ground truth detection achieves an IDF1 of 94.4%, which significantly outperforms that based on detections from the trained model. Hence, improving the sewer detection accuracy would be a priority to facilitating the tracking performance.

Second, as the metric learning model is utilized to extract features that are used in the tracking process, the quality of the metric learning model also influences the tracking results. Experiments demonstrate that the metric learning model performance in image matching and tracking is influenced by the training configurations i.e. loss functions and training dataset. The model trained with magnet loss converges at a faster speed and tracking based on features from this model achieves the highest IDF1 score, which indicates magnet loss is more suitable for the metric learning model in our experiment. In addition, we find that the model trained with target domain dataset outperforms using data from other domains. Meanwhile, combining training datasets from both the target domain and other domains further boosts the model performance, which probably is because the model learns more generic low-level features as well as discriminative features for re-identifying sewer defects. It is noticed that the metric

learning model with higher image matching accuracy tend to improve tracking, although the highest accuracy for image matching may not always generate the best tracking results. This phenomenon can be partially attributed to the more dynamic characteristics in the tracking process compared with relative static query process in the image matching task. In general, strategies such as different loss functions and multi-domain datasets can be applied to improve the tracking performance and to reduce the data preparation efforts.

Lastly, after investigating the influence of different weights of the three distance metrics, we find that assigning a larger weight to appearance and defect class distance tends to improve the tracking performance in our experiments. Inversely, larger weight to the motion distance may degrade the tracking. We also observe that tracking result is more sensitive to the object motion compared with appearance and defect class. Hence, it is important to find suitable weights for the distance metrics such as to improve the tracking performance. Generally, the observations in our experiments can provide insights for choosing appropriate weight values for the distance metrics in other studies. Meanwhile, it should be noted that the optimal weight values are obtained based on our sewer datasets, which may vary depending on the datasets used.

5. Conclusion

Despite the increasing trend of vision-based approaches for automated interpretation of sewer inspection videos, no studies have attempted to count the number of unique defects in the video, which is required for evaluating pipe condition. This paper proposed a framework to obtain defect number by tracking multiple sewer defects in CCTV videos based on defect detection and metric learning. First, a defect detection and a metric learning model based on deep learning is trained respectively using sewer defect data, such as to detect defects in each frame and extract their discriminative features. Then, based on three distance metrics, i.e. motion, appearance and defect type, the defect tracking module is proposed, which works by (1) detecting defects and extracting their features, (2) using Kalman filter to predict track state, (3) calculating three distance metrics between defects and tracks to obtain the cost matrix, and (4) associating detections with tracks based on the cost matrix.

Experiment results demonstrate that our framework achieves a decent performance with an IDF1 of 57.4% and performs well in tracking multiple defects under different camera motions (e.g. camera panning and rotation) as well as different illumination and blurry conditions, which can facilitate counting unique defect number. There are several findings from the experiments. First, the tracking performance greatly relies on the detection accuracy, indicating the importance of improving the detection model to further improve tracking results. Second, the metric learning model trained with different loss functions and datasets leads to different performance in both image matching and defect tracking. Training the model with datasets from multiple domains improves the model accuracy, which can greatly reduce the time and efforts for preparing target domain data. Third, there are different influences from the three proposed distance metrics on the tracking process. The tracking performance is more sensitive to the defect motion variation compared with the other two metrics. At the same time, different weights assigned to the distance metrics have different impacts, i.e. larger weights to appearance and defect class metrics generally improve the tracking while a larger weight to the motion distance probably decreases the accuracy.

Overall, this paper contributes by proposing a defect tracking framework based on defect detection, feature extraction and defect tracking using three distance metrics. To the knowledge of the authors, this is the first work attempting to track sewer defects in CCTV videos, which is expected to assist with counting unique defect number and evaluating pipe condition. Additionally, influences of each module are analyzed and the experiment findings can improve the defect tracking results, as well as provide insights for other research related to object tracking or metric learning. The main limitation of the paper is that the proposed framework requires inputs from the detection and metric learning models, which need to be trained and implemented separately to generate the detections and features. To make the framework works better for real-time tracking, we will explore a method to combine the detection, feature extraction and tracking into an end-to-end workflow.

References

- [1] EPA, Why control sanitary sewer overflows, 2001.
https://www3.epa.gov/npdes/pubs/sso_casestudy_control.pdf.
- [2] EPA, Report to congress on impacts and control of combined sewer overflows and sanitary sewer overflows, 2004. https://www.epa.gov/sites/production/files/2015-10/documents/csosortc2004_full.pdf.
- [3] ASCE, 2017 Wastewater infrastructure report card, 2017.
<https://www.infrastructurereportcard.org/wp-content/uploads/2017/01/Wastewater-Final.pdf> (accessed May 2, 2019).
- [4] NASSO, Pipeline Assessment (PACP), (2019).
<https://www.nassco.org/content/pipeline-assessment-pacp> (accessed June 11, 2019).
- [5] K. Wong, R.J. Allen, Utility Training Institute., Hong Kong conduit condition evaluation codes 2009 (HKCCEC2009) : code of practice for conduit condition evaluation using CCTV in Hong Kong, 4th ed., Advanced Technovation Ltd, Hong Kong, 2009. <https://www.worldcat.org/title/hong-kong-conduit-condition-evaluation-codes-2009-hkccec2009-code-of-practice-for-conduit-condition-evaluation-using-cctv-in-hong-kong/oclc/474643978> (accessed July 5, 2019).
- [6] S.S. Kumar, D.M. Abraham, M.R. Jahanshahi, T. Iseley, J. Starr, Automated defect classification in sewer closed circuit television inspections using deep convolutional neural networks, *Automation in Construction*. 91 (2018) 273–283.
doi:10.1016/j.autcon.2018.03.028.
- [7] S.S. Kumar, M. Wang, D.M. Abraham, M.R. Jahanshahi, T. Iseley, J.C.P. Cheng, Deep learning-based automated detection of sewer defects in CCTV videos, *Journal of Computing in Civil Engineering*. 34 (2020). doi:10.1061/(ASCE)CP.1943-5487.0000866.
- [8] J.C.P. Cheng, M. Wang, Automated detection of sewer pipe defects in closed-circuit television images using deep learning techniques, *Automation in Construction*. 95 (2018) 155–171. doi:10.1016/j.autcon.2018.08.006.
- [9] M. Wang, J.C.P. Cheng, A unified convolutional neural network integrated with conditional random field for pipe defect segmentation, *Computer-Aided Civil and Infrastructure Engineering*. (2019) mice.12481. doi:10.1111/mice.12481.
- [10] J.B. Haurum, T.B. Moeslund, A survey on image-based automation of CCTV and SSET sewer inspections, *Automation in Construction*. 111 (2020).
doi:10.1016/j.autcon.2019.103061.
- [11] T.C. Su, M. Der Yang, Application of morphological segmentation to leaking defect detection in sewer pipelines, *Sensors (Switzerland)*. 14 (2014) 8686–8704.

doi:10.3390/s140508686.

[12] M.R. Halfawy, J. Hengmeechai, Efficient algorithm for crack detection in sewer images from closed-circuit television inspections, *Journal of Infrastructure Systems*. 20 (2014) 04013014. doi:10.1061/(ASCE)IS.1943-555X.0000161.

[13] H. Ganegedara, D. Alahakoon, J. Mashford, A. Paplinski, K. Müller, T.M. Deserno, Self organising map based region of interest labelling for automated defect identification in large sewer pipe image collections, in: *Proceedings of the International Joint Conference on Neural Networks*, 2012: pp. 10–15. doi:10.1109/IJCNN.2012.6252482.

[14] W. Guo, L. Soibelman, J.H. Garrett, Automatic defect detection and recognition for asset condition assessment: A case study on sewer pipeline infrastructure system, in: *Congress on Computing in Civil Engineering, Proceedings*, 2007: pp. 419–426. doi:10.1061/40937(261)51.

[15] S. Iyer, S.K. Sinha, A robust approach for automatic detection and segmentation of cracks in underground pipeline images, *Image and Vision Computing*. 23 (2005) 921–933. doi:10.1016/j.imavis.2005.05.017.

[16] S. Iyer, S.K. Sinha, Segmentation of pipe images for crack detection in buried sewers, *Computer-Aided Civil and Infrastructure Engineering*. 21 (2006) 395–410. doi:10.1111/j.1467-8667.2006.00445.x.

[17] S.K. Sinha, P.W. Fieguth, Automated detection of cracks in buried concrete pipe images, *Automation in Construction*. 15 (2006) 58–72. doi:10.1016/j.autcon.2005.02.006.

[18] M. Der Yang, T.C. Su, Segmenting ideal morphologies of sewer pipe defects on CCTV images for automated diagnosis, *Expert Systems with Applications*. 36 (2009) 3562–3573. doi:10.1016/j.eswa.2008.02.006.

[19] N. Sarshar, M. Halfawy, J. Hengmeechai, Video processing techniques for assisted CCTV inspection and condition rating of sewers, *Journal of Water Management Modeling*. (2009) 235–8. doi:10.14796/jwmm.r235-08.

[20] M.R. Halfawy, J. Hengmeechai, Automated defect detection in sewer closed circuit television images using histograms of oriented gradients and support vector machine, *Automation in Construction*. 38 (2014) 1–13. doi:10.1016/j.autcon.2013.10.012.

[21] M.R. Halfawy, J. Hengmeechai, Integrated vision-based system for automated defect detection in sewer closed circuit television inspection videos, *Journal of Computing in Civil Engineering*. 29 (2015) 04014024. doi:10.1061/(ASCE)CP.1943-5487.0000312.

[22] S. Moradi, T. Zayed, F. Golkhoo, S. Moradi, T. Zayed, F. Golkhoo, Review on computer aided sewer pipeline defect detection and condition assessment, *Infrastructures*. 4 (2019) 10. doi:10.3390/infrastructures4010010.

[23] T.C. Su, M. Der Yang, T.C. Wu, J.Y. Lin, Morphological segmentation based on edge detection for sewer pipe defects on CCTV images, *Expert Systems with Applications*. 38 (2011) 13094–13114. doi:10.1016/j.eswa.2011.04.116.

[24] A. Hawari, M. Alamin, F. Alkadour, M. Elmasry, T. Zayed, Automated defect detection tool for closed circuit television (CCTV) inspected sewer pipelines, *Automation in Construction*. 89 (2018) 99–109. doi:10.1016/j.autcon.2018.01.004.

[25] M. Kolesnik, G. Barattoff, 3-D interpretation of sewer circular structures, *Proceedings - IEEE International Conference on Robotics and Automation*. 2 (2000) 1453–1458. doi:10.1109/robot.2000.844802.

[26] X. Pan, T.A. Clarke, T.J. Ellis, Detection and tracking of pipe joints in noisy images, in: *Proceedings - SPIE 2350, Videometrics III*, 1994: pp. 136–147. doi:10.1117/12.189127.

[27] W. Guo, L. Soibelman, J.H. Garrett, Automated defect detection in urban wastewater

- 1161 pipes using invariant features found in video images, in: Construction Research
1162 Congress 2009, Seattle, United States, 2009: pp. 1194–1203.
1163 doi:10.1061/41020(339)121.
- 1164 [28] L.M. Dang, S.I. Hassan, S. Im, I. Mehmood, H. Moon, Utilizing text recognition for
1165 the defects extraction in sewers CCTV inspection videos, *Computers in Industry*. 99
1166 (2018) 96–109. doi:10.1016/j.compind.2018.03.020.
- 1167 [29] J. Myrans, Z. Kapelan, R. Everson, Automated detection of faults in wastewater pipes
1168 from CCTV footage by using random forests, in: *Procedia Engineering*, 2016: pp. 36–
1169 41. doi:10.1016/j.proeng.2016.07.416.
- 1170 [30] J. Myrans, R. Everson, Z. Kapelan, Automated detection of faults in sewers using
1171 CCTV image sequences, *Automation in Construction*. 95 (2018) 64–71.
1172 doi:10.1016/j.autcon.2018.08.005.
- 1173 [31] G. Heo, J. Jeon, B. Son, Crack automatic detection of CCTV video of sewer inspection
1174 with low resolution, *KSCE Journal of Civil Engineering*. 23 (2019) 1219–1227.
1175 doi:10.1007/s12205-019-0980-7.
- 1176 [32] W. Guo, L. Soibelman, J.H. Garrett, Visual pattern recognition supporting defect
1177 reporting and condition assessment of wastewater collection systems, *Journal of*
1178 *Computing in Civil Engineering*. 23 (2009) 160–169. doi:10.1061/ASCE0887-
1179 3801200923:3160.
- 1180 [33] W. Guo, L. Soibelman, J.H. Garrett, Automated defect detection for sewer pipeline
1181 inspection and condition assessment, *Automation in Construction*. 18 (2009) 587–596.
1182 doi:10.1016/j.autcon.2008.12.003.
- 1183 [34] M.R. Halfawy, J. Hengmeechai, Optical flow techniques for estimation of camera
1184 motion parameters in sewer closed circuit television inspection videos, *Automation in*
1185 *Construction*. 38 (2014) 39–45. doi:10.1016/j.autcon.2013.10.016.
- 1186 [35] O. Moselhi, T. Shehab-Eldeen, Classification of defects in sewer pipes using neural
1187 networks, *Journal of Infrastructure Systems*. 6 (2000) 97–104.
1188 doi:10.1061/(ASCE)1076-0342(2000)6:3(97).
- 1189 [36] T. Shehab, O. Moselhi, Automated detection and classification of infiltration in sewer
1190 pipes, *Journal of Infrastructure Systems*. 11 (2005) 165–171.
1191 doi:10.1061/(ASCE)1076-0342(2005)11:3(165).
- 1192 [37] M.D. Yang, T.C. Su, Automated diagnosis of sewer pipe defects based on machine
1193 learning approaches, *Expert Systems with Applications*. 35 (2008) 1327–1337.
1194 doi:10.1016/j.eswa.2007.08.013.
- 1195 [38] M.J. Chae, D.M. Abraham, Neuro-fuzzy approaches for sanitary sewer pipeline
1196 condition assessment, *Journal of Computing in Civil Engineering*. 15 (2002) 4–14.
1197 doi:10.1061/(asce)0887-3801(2001)15:1(4).
- 1198 [39] S.K. Sinha, M.A. Knight, Intelligent system for condition monitoring of underground
1199 pipelines, *Computer-Aided Civil and Infrastructure Engineering*. 19 (2004) 42–53.
1200 doi:10.1111/j.1467-8667.2004.00336.x.
- 1201 [40] D. Bairaktaris, V. Delis, C. Emmanouilidis, S. Frondistou-Yannas, K. Gratsias, V.
1202 Kallidromitis, N. Rerras, Decision-support system for the rehabilitation of
1203 deteriorating sewers, *Journal of Performance of Constructed Facilities*. 21 (2007) 240–
1204 248. doi:10.1061/(ASCE)0887-3828(2007)21:3(240).
- 1205 [41] J. Mashford, M. Rahilly, P. Davis, S. Burn, A morphological approach to pipe image
1206 interpretation based on segmentation by support vector machine, *Automation in*
1207 *Construction*. 19 (2010) 875–883. doi:10.1016/j.autcon.2010.06.001.
- 1208 [42] J. Myrans, R. Everson, Z. Kapelan, Automated detection of fault types in CCTV sewer
1209 surveys, *Journal of Hydroinformatics*. 21 (2019) 153–163.
1210 doi:10.2166/hydro.2018.073.

- 1211 [43] S.K. Sinha, P.W. Fieguth, Neuro-fuzzy network for the classification of buried pipe
1212 defects, *Automation in Construction*. 15 (2006) 73–83.
1213 doi:10.1016/j.autcon.2005.02.005.
- 1214 [44] S.K. Sinha, P.W. Fieguth, M.A. Polak, Computer vision techniques for automatic
1215 structural assessment of underground pipes, *Computer-Aided Civil and Infrastructure*
1216 *Engineering*. 18 (2003) 95–112. doi:10.1111/1467-8667.00302.
- 1217 [45] S.I. Hassan, L.M. Dang, I. Mehmood, S. Im, C. Choi, J. Kang, Y.-S. Park, H. Moon,
1218 Underground sewer pipe condition assessment based on convolutional neural
1219 networks, *Automation in Construction*. 106 (2019) 102849.
1220 doi:10.1016/j.autcon.2019.102849.
- 1221 [46] Q. Xie, D. Li, J. Xu, Z. Yu, J. Wang, Automatic detection and classification of sewer
1222 defects via hierarchical deep learning, *IEEE Transactions on Automation Science and*
1223 *Engineering*. 16 (2019) 1836–1847. doi:10.1109/TASE.2019.2900170.
- 1224 [47] X. Yin, Y. Chen, A. Bouferguene, H. Zaman, M. Al-Hussein, L. Kurach, A deep
1225 learning-based framework for an automated defect detection system for sewer pipes,
1226 *Automation in Construction*. 109 (2020) 102967. doi:10.1016/j.autcon.2019.102967.
- 1227 [48] D. Li, A. Cong, S. Guo, Sewer damage detection from imbalanced CCTV inspection
1228 data using deep convolutional neural networks with hierarchical classification,
1229 *Automation in Construction*. 101 (2019) 199–208. doi:10.1016/j.autcon.2019.01.017.
- 1230 [49] J. Künzel, T. Werner, P. Eisert, J. Waschnewski, R. Möller, R. Hilpert, Automatic
1231 analysis of sewer pipes based on unrolled monocular fisheye images, in: *Proceedings -*
1232 *2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018,*
1233 *2018: pp. 2019–2027.* doi:10.1109/WACV.2018.00223.
- 1234 [50] G. Ciaparrone, F. Luque Sánchez, S. Tabik, L. Troiano, R. Tagliaferri, F. Herrera,
1235 Deep learning in video multi-object tracking: A survey, *Neurocomputing*. 381 (2020)
1236 61–88. doi:10.1016/j.neucom.2019.11.023.
- 1237 [51] D.S. Bolme, J.R. Beveridge, B.A. Draper, Y.M. Lui, Visual object tracking using
1238 adaptive correlation filters, in: *Proceedings of the IEEE Computer Society Conference*
1239 *on Computer Vision and Pattern Recognition, IEEE, 2010: pp. 2544–2550.*
1240 doi:10.1109/CVPR.2010.5539960.
- 1241 [52] J.F. Henriques, R. Caseiro, P. Martins, J. Batista, High-speed tracking with kernelized
1242 correlation filters, *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
1243 37 (2015) 583–596. doi:10.1109/TPAMI.2014.2345390.
- 1244 [53] C. Xiao, A. Yilmaz, Efficient tracking with distinctive target colors and silhouette, in:
1245 *Proceedings - International Conference on Pattern Recognition, Institute of Electrical*
1246 *and Electronics Engineers Inc., 2016: pp. 2728–2733.*
1247 doi:10.1109/ICPR.2016.7900048.
- 1248 [54] G. Boudoukh, I. Leichter, E. Rivlin, Visual tracking of object silhouettes, in:
1249 *Proceedings - International Conference on Image Processing, ICIP, IEEE Computer*
1250 *Society, 2009: pp. 3625–3628.* doi:10.1109/ICIP.2009.5414280.
- 1251 [55] V. Bruni, D. Vitulano, An improvement of kernel-based object tracking based on
1252 human perception, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. 44
1253 (2014) 1474–1485. doi:10.1109/TSMC.2014.2331217.
- 1254 [56] W. Chen, B. Niu, H. Gu, X. Zhang, A novel strategy for kernel-based small target
1255 tracking against varying illumination with multiple features fusion, in: *2018*
1256 *International Conference on Information and Computer Technologies, ICICT 2018,*
1257 *Institute of Electrical and Electronics Engineers Inc., 2018: pp. 135–138.*
1258 doi:10.1109/INFOCT.2018.8356856.
- 1259 [57] M. Ighrayene, G. Qiang, T. Benlefi, Making Bayesian tracking and matching by the
1260 BRISK interest points detector/descriptor cooperate for robust object tracking, in: *2016*

1261 IEEE International Conference on Signal and Image Processing, ICSIP 2016, Institute
1262 of Electrical and Electronics Engineers Inc., 2017: pp. 731–735.
1263 doi:10.1109/SIPROCESS.2016.7888360.

1264 [58] S.M. Marvasti-Zadeh, L. Cheng, H. Ghanei-Yakhdan, S. Kasaei, Deep learning for
1265 visual tracking: a comprehensive survey, ArXiv. abs/1912.0 (2019).
1266 <http://arxiv.org/abs/1912.00535> (accessed April 2, 2020).

1267 [59] A. Bewley, Z. Ge, L. Ott, F. Ramos, B. Upcroft, Simple online and realtime tracking,
1268 in: Proceedings - International Conference on Image Processing, ICIP, 2016: pp.
1269 3464–3468. doi:10.1109/ICIP.2016.7533003.

1270 [60] N. Mahmoudi, S.M. Ahadi, M. Rahmati, Multi-target tracking using CNN-based
1271 features: CNNMTT, Multimedia Tools and Applications. 78 (2019) 7077–7096.
1272 doi:10.1007/s11042-018-6467-6.

1273 [61] N. Wojke, A. Bewley, D. Paulus, Simple online and realtime tracking with a deep
1274 association metric, in: Proceedings - International Conference on Image Processing,
1275 ICIP, 2018: pp. 3645–3649. doi:10.1109/ICIP.2017.8296962.

1276 [62] M.W. Park, A. Makhmalbaf, I. Brilakis, Comparative study of vision tracking methods
1277 for tracking of construction site resources, Automation in Construction. 20 (2011)
1278 905–915. doi:10.1016/j.autcon.2011.03.007.

1279 [63] J. Yang, O. Arif, P.A. Vela, J. Teizer, Z. Shi, Tracking multiple workers on
1280 construction sites using video cameras, in: Advanced Engineering Informatics,
1281 Elsevier, 2010: pp. 428–434. doi:10.1016/j.aei.2010.06.008.

1282 [64] E. Konstantinou, J. Lasenby, I. Brilakis, Adaptive computer vision-based 2D tracking
1283 of workers in complex environments, Automation in Construction. 103 (2019) 168–
1284 184. doi:10.1016/j.autcon.2019.01.018.

1285 [65] M.W. Park, I. Brilakis, Continuous localization of construction workers via integration
1286 of detection and tracking, Automation in Construction. 72 (2016) 129–142.
1287 doi:10.1016/j.autcon.2016.08.039.

1288 [66] E. Rezazadeh Azar, S. Dickinson, B. McCabe, Server-customer interaction tracker:
1289 Computer vision-based system to estimate dirt-loading cycles, Journal of Construction
1290 Engineering and Management. 139 (2013) 785–794. doi:10.1061/(ASCE)CO.1943-
1291 7862.0000652.

1292 [67] Z. Zhu, X. Ren, Z. Chen, Integrated detection and tracking of workforce and
1293 equipment from construction jobsite videos, Automation in Construction. 81 (2017)
1294 161–171. doi:10.1016/j.autcon.2017.05.005.

1295 [68] I. Brilakis, M.W. Park, G. Jog, Automated vision tracking of project related entities,
1296 Advanced Engineering Informatics. 25 (2011) 713–724.
1297 doi:10.1016/j.aei.2011.01.003.

1298 [69] Y.J. Lee, M.W. Park, 3D tracking of multiple onsite workers based on stereo vision,
1299 Automation in Construction. 98 (2019) 146–159. doi:10.1016/j.autcon.2018.11.017.

1300 [70] D. Roberts, M. Golparvar-Fard, End-to-end vision-based detection, tracking and
1301 activity analysis of earthmoving equipment filmed at ground level, Automation in
1302 Construction. 105 (2019). doi:10.1016/j.autcon.2019.04.006.

1303 [71] O. Angah, A.Y. Chen, Tracking multiple construction workers through deep learning
1304 and the gradient based method with re-matching based on multi-object tracking
1305 accuracy, Automation in Construction. 119 (2020). doi:10.1016/j.autcon.2020.103308.

1306 [72] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection
1307 with region proposal networks, IEEE Transactions on Pattern Analysis and Machine
1308 Intelligence. 39 (2017) 1137–1149. doi:10.1109/TPAMI.2016.2577031.

1309 [73] S.S. Kumar, Leveraging big-data and deep learning for economical condition
1310 assessment of wastewater pipelines, Purdue University Graduate School, 2020.

- doi:10.25394/PGS.12218432.V1.
- [74] N. Wojke, A. Bewley, Deep cosine metric learning for person re-identification, Proceedings - 2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018. 2018-Janua (2018) 748–756. doi:10.1109/WACV.2018.00087.
 - [75] K. Zhou, Y. Yang, A. Cavallaro, T. Xiang, Omni-scale feature learning for person re-identification, in: Proceedings of the IEEE International Conference on Computer Vision, 2019: pp. 3701–3711. doi:10.1109/ICCV.2019.00380.
 - [76] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, Q. Tian, Scalable person re-identification: A benchmark, in: Proceedings of the IEEE International Conference on Computer Vision, 2015: pp. 1116–1124. doi:10.1109/ICCV.2015.133.
 - [77] W. Li, R. Zhao, T. Xiao, X. Wang, DeepReID: Deep filter pairing neural network for person re-identification, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2014: pp. 152–159. doi:10.1109/CVPR.2014.27.
 - [78] E. Ristani, F. Solera, R. Zou, R. Cucchiara, C. Tomasi, Performance measures and a data set for multi-target, multi-camera tracking, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2016: pp. 17–35. doi:10.1007/978-3-319-48881-3_2.
 - [79] Mahmut Kaya, Hasan Sakir Bilge, Deep metric learning : A survey, Symmetry. 1066 (2019). doi:10.3390/sym11091066.
 - [80] K. Bernardin, R. Stiefelhagen, Evaluating multiple object tracking performance: The CLEAR MOT metrics, Eurasip Journal on Image and Video Processing. 2008 (2008). doi:10.1155/2008/246309.
 - [81] Tzutalin, LabelImg, (2015). <https://github.com/tzutalin/labelImg> (accessed July 14, 2019).
 - [82] C. Vondrick, D. Patterson, D. Ramanan, Efficiently scaling up crowdsourced video annotation: A set of best practices for high quality, economical video labeling, International Journal of Computer Vision. 101 (2013) 184–204. doi:10.1007/s11263-012-0564-1.
 - [83] E. Hoffer, N. Ailon, Deep metric learning using triplet network, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2015: pp. 84–92. doi:10.1007/978-3-319-24261-3_7.
 - [84] O. Rippel, M. Paluri, P. Dollar, L. Bourdev, Metric learning with adaptive density discrimination, 4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings. (2016) 1–15. <https://arxiv.org/pdf/1511.05939.pdf>.