Pothole Detection Using Computer Vision and Learning

Amita Dhiman and Reinhard Klette

***Abstract*— Techniques for identifying potholes on road surfaces aim at developing strategies for real-time or offline identification of potholes, to support real-time control of a vehicle (for driver assistance or autonomous driving) or offline data collection for road maintenance. For these reasons, research around the world has comprehensively explored strategies for the identification of potholes on roads. This paper starts with a brief review of the field; it classifies developed strategies into several categories. We, then, present our contributions to this field by implementing strategies for automatic identification of potholes. We developed and studied two techniques based on stereo-vision analysis of road environments ahead of the vehicle; we also designed two models for deep-learning-based pothole detection. An experimental evaluation of those four designed methods is provided, and conclusions are drawn about particular benefits of these methods.**

***Index Terms*— Pothole detection, stereo vision, deep learning.**

I. Introduction

DISTRACTED driving, speeding or other driver errors are main causes of accidents worldwide; however, bad road conditions are also a significant cause. The condition of a road turns out to be dangerous due to number of reasons such as flooding, rain, damages caused, e.g., by overloaded big vehicles, or poor physical maintenance of the road. Road condition assessment involves identifying and analyzing distinct types of road surface distress, like potholes, cracks or texture changes as being maintenance-relevant features. *Macro-scale* road features are defined by being of traffic relevance. For example, speed bumps are also traffic-relevant features; they also require detection for driver assistance.

A *pothole* is a special case of road distress. It can be an arbitrarily shaped structural defect of a road, and a precise identification of its "border" is typically impossible; see Fig. 1. They can be vaguely outlined, but their maximum depth can be identified more precisely. Objects such as cars, persons, cyclists, dogs or cats are of specifically defined shapes (and now detected by deep learning due to appearance properties); compared to this, we can certainly claim that the detection of a pothole, being of arbitrary shape and of complex geometric structure, is a challenging object-detection task.

Potholes present a grave danger to human life. We just state a few facts from related studies worldwide. According to a Chicago Sun-Times analysis of city data, drivers filed 11,706 complaints about potholes with the city in the first two months of 2018 [1]. In the UK, about 50 cyclists are seriously injured every year because of Britain's poor roads [2]. In India, 3, 597 people died due to potholes [3].

Fig. 1. Complex geometric shape of a pothole not supporting a precise definition of a "border".

Potholes may cause significant costs. For example, in 2017, different city councils in New Zealand have spend the following in order to fix potholes: Christchurch 525, 000, Wellington 12, 782, Invercargill 60, 000, and Dunedin 27, 000; see [4].

Extensive research has been carried out for macro-scale road issues, such as for estimating the road surface [5] (also known as *road manifold* estimation), detection of obstacles that are protruding from the road [6], recognition of traffic isles [7], or pothole detection [8]. Automotive companies such as Tesla, Toyota, Ford, or BMW announced to be able to deliver autonomous cars by about 2020 [9]. However, road pothole detection as a particular research subject still demands more research (as, certainly, many more related topics in this area).

Mobile crowdsourcing based applications have been developed to report about road hazards, such as Santani *et al.* [12]. In 2017, a study conducted in Taoyuan, Taiwan, used a data-analytic approach applying correlation and regression analysis; [11] shows that areas, identified (by crowdsourcing) for having a high frequency of road potholes, resulted in a higher number of traffic accidents. In 2018, one of the largest pizza chains in the U.S., dispensed a special grant to fix potholes at selected locations, as potholes caused irreversible damage to pizzas during their delivery [13].

Authors published already about three developed methods for pothole detection in conference papers [14]–[16]. This paper builds on those reported materials. This paper provides at first a review on techniques for pothole identification, extending brief notes on related literature in those previous

Manuscript received March 6, 2019; revised May 25, 2019; accepted July 12, 2019. The work of A. Dhiman was supported by a Ph.D. Scholarship from Aukland University of Technology. The Associate Editor for this paper was D. F. Wolf. *(Corresponding author: Reinhard Klette.)*

The authors are with the School of Engineering, Computer and Mathematical Sciences, EEE Department, Auckland University of Technology, Auckland 1010, New Zealand (e-mail: amita.dhiman@aut.ac.nz; reinhard.klette@aut.ac.nz).

Digital Object Identifier 10.1109/TITS.2019.2931297

conference papers. This paper also presents (with additional material) the three previously published methods, adds one more method, and provides a comparative evaluation of all four methods. For this evaluation, we use here (first time) a more diverse set of data. Potholes presents different challenges under different weather, lighting, road geometry or traffic conditions. As there is no online benchmark dataset available for pothole detection, we accumulated data from multiple sources, and suggest to use those five different datasets, recorded under different weather conditions, for future discussions of progress in this field of pothole detection.

Contributions of this study are two different approaches of pothole detection based either on 3D scene reconstruction or on state-of-the-art deep learning techniques. The proposed strategies allow us identifications of potholes from a distance in an accurate manner as supported by experiments. Evaluated experiments demonstrate that state-of-the-art deep learning-based methods significantly outperform the conventional 3D scene reconstruction-based methods. We believe that this study sheds new lights into the field of pothole detection by bridging a gap caused by datasets recorded under varying illumination conditions. This study is also an overview of previously conducted attempts to detect defects on road surface.

This paper is structured as follows. Section II provides a review of reported work on pothole detection. Section II-A describes manual techniques, i.e. techniques which use a human as sensor for the detection of road anomalies. Section II-B reviews techniques that use accelerometers or gyroscopes as vibration detection systems to measure vibrations that turn out in a vehicle whenever it strikes any distress on a road. Section II-C presents basic strategies of implying image or video processing techniques for road distress detection. Section II-D reviews techniques based on *3-dimensional* (3D) scene reconstruction. Section II-E lists current work using learning strategies. Section III informs about methods proposed by the authors. This starts in Section III-A with a technique using single stereo-frame data, further improved in Section III-B by the use of multi-frame stereo data based on visual odometry. Sections III-C and III-D address then learning-based strategies, first by using transfer learning method based on Mask R-CNN, then by using transfer learning using YOLOv2. Section IV gives information about the datasets used for comparative performance evaluation. Section V details the studied experiments. Section VI concludes.

II. Literature Review

This review might contribute to the motivation of developing automated road-surface anomaly detection systems for various real-world environments. There have been much advancements in this technical era recently.

A. Public Reporting

This type of systems enhances civic engagements by government, and facilitates the participation by citizens of the country. These systems use the public as sensors [10]. The main advantage of this method is that there is no need for costly hardware or software. Citizens can report a pothole by capturing its picture with their mobile devices and later by uploading or sending to a website or application, or by merely sending information about a pothole's location. Some reported systems are listed in Table I.

TABLE I

Public Reporting; Listed Names on the Left Identify the Websites of Those Applications (e.g. `www.fixmystreet.com`). Citizen Hotline 1999 is the Name of an Innovative Open Data Platform Used in Taiwan; the Related Research Publication Was in 2017

App/Website	Countries
FixMyStreet	UK, New Zealand
SeeClickFix	US
Citizens-Connect	Netherlands, Canada
PDX Reporter	Portland
Report a Pothole	London
BBMP	Bangalore, India
Citizen Hotline 1999	Taoyuan, Taiwan

TABLE II

Vibration-Based Methods

Authors	Years	Sensors
Jintin Ren et al.	2017	Accelerometer, GPS
Fatjon Seraj et al.	2016	Accelerometer, gyroscope and GPS
Marcin Badurowicz et al.	2016	Smartphone, Accelerometer
Chih-Wei et al.	2015	Accelerometer, Smartphone
Manjusha Ghadge et al.	2015	Smartphone, Accelerometer,
Mohamed Fekry et al.	2014	Accelerometer and GPS
Artis Mednis et al.	2011	Accelerometer, Smartphone, Laptop
Thegaran Naidoo et al.	2011	Accelerometer and GPS

B. Vibration-Based Methods

Vibration-based methods include approaches of collecting abnormal vibrations [17] caused in the vehicles while driving over road anomalies. Vibrations of the vehicle are collected using an accelerometer; see Table II. The main drawback of the vibration-based methods is that the vehicle has to drive over the pothole in order to measure the vibrations caused by the pothole on the road.

M. Ghadge *et al.* [18] used an accelerometer and GPS to analyze the conditions of roads to detect locations of potholes and bumps using a machine learning approach, defined by K-means clustering on training data and a random-forest classifier for testing data. The data is divided first into two clusters of "pothole" or "non-pothole", and then a random-forest classifier is used to validate the proposed result provided by the clustering algorithm. It is reported that clustering does not perform well when clusters of different size and severity are involved; size and severity of a pothole are the major properties considered in the system.

F. Seraj *et al.* [19] used a support vector machine (SVM) for a machine-learning approach to classify road anomalies. The proposed system uses accelerometer, gyroscope and a Samsung galaxy as sensors for data collection; data labeling is performed manually (by a human) and then a high-pass filter is used to remove the low-frequency components caused due to turns and accelerations. Ren *et al.* [20] used K-means clustering to detect potholes based on data collected

by using an accelerometer and GPS. The proposed system lacks accuracy regarding the isolation of potholes from other road anomalies.

C. 2D-Vision-Based Methods

Vision-based methods use *2-dimensional* (2D) image or video data, captured using a digital camera, and process this data using 2D images or video processing techniques [21], [22]. The choice of the applied image processing techniques is highly dependent on the application for which 2D images are being processed.

Koch and Brilakis [8] proposed a method aiming at a separation of defect and non-defect regions in an image using histogram shape based threshold. The authors consider the shape of a pothole as being approximately elliptical based on a perspective view. The authors emphasize on using machine learning in future work, and claim that the proposed work already results in 86% `Accuracy` along with 86% `Recall` and 82% `Precision`, with the common definitions of

$$\texttt{Precision} = \frac{\texttt{TP}}{\texttt{TP}+\texttt{FP}} \tag{1}$$

$$\texttt{Recall} = \frac{\texttt{TP}}{\texttt{TP}+\texttt{FN}} \tag{2}$$

$$\texttt{Accuracy} = \frac{\texttt{TP}+\texttt{TN}}{\texttt{TP}+\texttt{TN}+\texttt{FP}+\texttt{FN}} \tag{3}$$

$$\texttt{F1} = 2 \cdot \frac{\texttt{Precision} \cdot \texttt{Recall}}{\texttt{Precision}+\texttt{Recall}} \tag{4}$$

where `TP` is the number of true positives, `FP` of false positives, `TN` of true negatives, and `FN` of false negatives.

Tedeschi and Benedetto [10] recently suggested a system for *automatic pavement distress recognition* (`ADPR`) which is able to perform in real time by identifying road distress including fatigue cracks, longitudinal and traversal cracks, and potholes. The authors used a combination of technologies of the OpenCV library and for the classification of the three different types of road distresses, three classifiers have been used based on *local binary pattern* (LBP) features; they achieved more than 70% for `Precision`, `Recall`, and the `F1`-measure.

Authors discussed difficulties of defining the severity of considered kinds of road distresses. For texture classification the authors used Haralick's features [23] based on *gray level co-occurrence matrices* (GLCMs) and then classified image regions using a tool from [24].

Ryu *et al.* [26] proposed a method to detect potholes both for asphalt or concrete road surfaces using 2D images collected by a mounted optical device on a survey vehicle. The system mainly works in three steps of image segmentation, candidate region extraction and decision. The system fails to detect potholes in darker images (image regions) due to shadows (e.g. of trees or cars) present in real-world road recordings.

Powell and Satheeshkumar [27] present a method for the detection of potholes by segmenting images into defected or non-defected regions. After extracting the texture information from defected regions, this texture information is compared with texture information obtained from non-defected regions. The proposed system considers shadow effects on the road and aims to remove those effects of shadows using a shadow-removal algorithm. The system is unable to perform in rainy weather. The authors concluded that the system should be further extended to perform also on video data as the system was only tested on 2D images collected using an iPhone camera with 5 megapixel image resolution.

Bashkar and Manohar [44] propose a methodology of detecting pothole's mean depth by using SURF features on uncalibrated stereo pairs of images (without employing disparity images). A particular methodology has been developed for this purpose, but appears to suffers from uncalibrated stereo rectification; it is far from providing good results.

Ying *et al.* [46] proposed a system which can detect road surface based on a feature detector which is shadow-occurence optimized. This system uses a connected-component-analysis algorithm and other morphological algorithms and is demonstrated on images of datasets provided by KITTI [47] and ROMA [48].

Thekkethala *et al.* [25] used two (stereoscopic) cameras and applied stereo matching to estimate depth of a pothole on asphalt pavement surface. After performing binarization and morphological operations, a skeleton of a pothole is estimated. The system is tested on 24 images and no estimates of depth have been provided. The system can detect skeletons of potholes of great depression. Authors did not estimate the road manifold.

D. 3D Scene Reconstruction-Based Methods

3D scene reconstruction is the method of capturing the shape, depth, and appearance of objects in the real world; it relies on 3D surface reconstruction which typically demands more computations than 2D vision. Rendering of surface elevations helps to understand accuracy during the design of 3D vision systems. 3D scene reconstruction can be based on using various types of sensors, such as Kinect [28], stereo-vision cameras, or a 3D laser. Kinect sensors are mainly used in fields of (indoor) robotics or gaming.

3D lasers define an advanced road-survey technology; compared to camera-based systems it still comes with higher costs; [30], [31] report survey cycles of (usually) once in four years. A 3D laser uses a laser source to illuminate the surface and a scan camera for capturing the created light patterns. [32] applied the common laser-line projection; the recorded laser line deforms when it strikes an obstacle (and supports thus the 3D reconstruction), but does not work well, e.g., on wet roads or potholes filled with water.

Stereo-vision cameras are considered to be cost-effective as compared to other sensors. Stereo vision aims at effective and accurate disparities, calculated from left-and-right image pairs, to be used for estimating depth or distance; see, for example, [33]. Commonly, the canonical left-right calibrated stereo camera setup is used while aiming at a reconstruction dense 3D surfaces. A *disparity map* represents per-pixel correspondences for a rectified stereo pair. Figure 2 illustrates a recorded 3D scene with a calculated (color-encoded) disparity map.

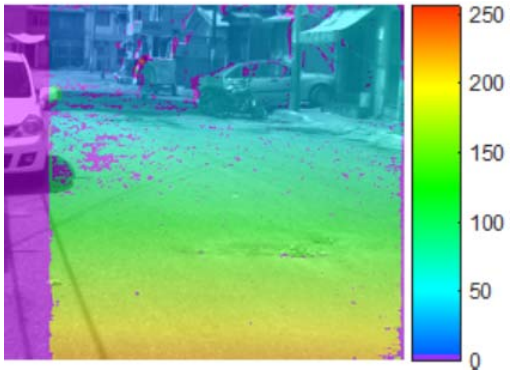

Fig. 2. Recorded road scene with transparent color-encoded disparity map calculated using stereo global matching algorithm [29]. The used color key is shown on the right; disparity 250 encodes a distance very close to the host vehicle and 0 encodes "very far away".

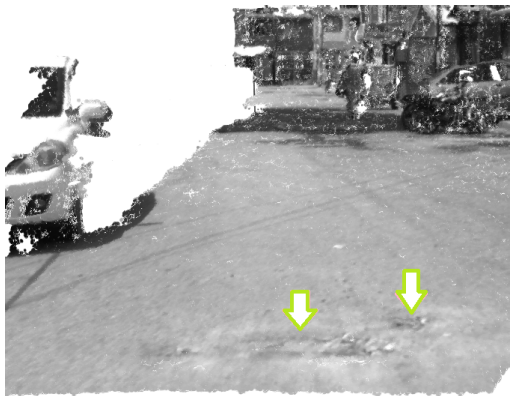

Fig. 3. Reconstructed (and texture-mapped) 3D point cloud using calculated disparities. Green arrows indicate locations of two potholes.

For briefly introducing the stereo-vision notation (later needed in this paper), consider coordinates $(x, y) \in \Omega$ of a pixel in an image, where Ω denotes the image domain of the left image. A disparity map $D : \Omega \to \mathbb{R}_0^+$ defines the translation of image coordinates in the left image into those of a detected corresponding pixels $(x - D(x, y), y)$ in the right image.

A 3D point (X, Y, Z) in the 3D scene is mapped into an image pixel at (x, y) following a perspective projection

$$x = f_x \cdot \frac{X}{Z} + x_c, \quad y = f_y \cdot \frac{Y}{Z} + y_c \tag{5}$$

where f_x and f_y are the focal lengths in x and y coordinate direction, and (x_c, y_c) is the principal point in the image plane.

Given a calibrated and rectified stereo camera pair, let $d = D(x, y)$ be the disparity value assigned to a left-image pixel location (x, y), the Z-coordinate can be triangulated as follows:

$$Z = f_x \cdot \frac{b}{d} \tag{6}$$

Here, b is the length of the baseline which is the distance between left and right camera optical centers. This allows us that X and Y can also be recovered; the whole process is called *triangulation*. Figure 3 shows a cloud of 3D points, reconstructed from a disparity map, following the triangulation process.

TABLE III
EXAMPLES OF 3D RECONSTRUCTION-BASED METHODS AND USED SENSORS

Authors	Years	Sensors
Tomasz Garbowski et al.	2017	Stereo-vision cameras
Vijaya Bashkar et al.	2016	Stereo-vision cameras
Aliaksei Mikhailiuk et al.	2016	MicroController TMS320C6678 DSP
A. Rasheed et al.	2015	Kinect sensors
Marcin Staniek	2015	Stereo-vision cameras, GPS and vibration sensor
Kiran Kumar Vupparaboina et al.	2015	Laser, camera
Zheng Zhang et al.	2014	Stereo-vision PointGrey Flea 3 cameras
He Youguan et al.	2011	Stereo-vision cameras and LED
X.Yu et al.	2011	Laser, camera

Table III summarizes a few 3D reconstruction-based methods for detecting road distress.

Garbowski and Gajewski [41] presented a semi-automatic *pavement failure detection system* (PFDS) which is a part of the FEMat [42] road package (UDPhoto toolbox). It allows a user to inspect the condition of road pavement based on calculated clouds of 3D points. The presented system considers a small *region of interest* (ROI) in reference to a larger region of a road surface and is able to detect certain types of cracks including "alligator cracks", but not potholes.

Shen *et al.* [43] propose the use of Takata's stereo-vision system for performing a road surface preview along the host vehicle. Video data recorded with the used compact stereo-vision sensor (with a baseline of 16.5 cm) is analyzed in an embedded system, already tested in various vehicles, also in combination with various driver assistance systems such as *forward collision warning* (FCW), *automatic emergency breaking* (AEB), or *lane departure warning* (LDW). Authors state that the proposed system achieves satisfactory accuracy; they also state that it does not perform well when there is glare on the road surface.

Calculated disparities within detected road-surface image segments support the estimation of a *manifold*, approximating the road-surface. Commonly the road surface is assumed to be planar (i.e., the manifold is thus a plane). But this planarity assumption is often not corresponding to actual uneven road surfaces. To simplify, the road manifold is often modeled in driving direction by a *profile*, i.e. a curve whose parallel translation left-to-right creates the road manifold. A line creates a plane, and a quadratic polynomial profile creates a quadratic road manifold.

Quadratic road manifolds are discussed by Ai *et al.* [6]. For a consideration of twisting and bending surfaces of roads, see Mikhailiuk and Dahnoun [55]; their algorithm has been implemented on a Texas Instrument C6678 multi-core SoC digital signal processor.

Zhang *et al.* [34] proposed an efficient algorithm to estimate the size, depth, position and severity of potholes by modeling the road surface as a quadratic manifold by using a *random sample census* (RANSAC) approach. Pothole detection and

TABLE IV
EXAMPLES OF CNNS FOR IMAGE SEGMENTATION, AND USED DATA SETS

CNN	Year	Used datasets
FCN	2014	PASCAL VOC
$SEGNET$	2015	CamVid
$DilatedConvolutions$	2015	VOC2012, COCO
$DeepLab$	2014-2017	PASCAL 2012, CityScapes
$RefineNet$	2016	PASCAL 2012
$PSPNET$	2016	PASCAL 2012, CityScapes
$LargeKernelMatters$	2017	PASCAL 2012, CityScapes
$MaskR-CNN$	2017	COCO

segmentation are achieved by using a *connected-component labeling* (CCL) algorithm.

Proposed methods may also follow a multi-sensor approach [36]–[40]. Tseng *et al.* [45] developed an automated survey robot which performed in simulated test-field environments to detect five types of distress, namely alligator cracks, small patches, potholes, rectangular, and circular manhole covers.

E. Learning-Based Methods

For identifying objects in image data, various *convolutional neural networks* (CNNs) have been proposed, see Table IV, such as Chen *et al.* [56], RefineNet [57], PSPNet [58], or the "large-kernel-matters" proposal in [59]. For identifying an object at pixel level, *fully convolutional neural networks* (FCNs) have been proposed; for example, see Long *et al.* [60], or SegNet by Badrinaraynan *et al.* [61].

Regarding road damage detection, Zhang *et al.* [67] propose a CrackNet to predict class scores for all the pixels in a considered damage. Song *et al.* [62] use a CNN approach to detect potholes. The authors used a smartphone as a sensor to acquire movement information and the Inception V3 [63] classifier; they adapted the final fully-connected layer in the CNN to the given task.

Maeda *et al.* [64] used a CNN, trained by using a vast dataset of road images collected in Japan, to detect road-surface damage. Some authors used SSD Inception V2 [65] or SSD MobileNet [66] to identify different sorts of road damages. Detected road damages are identified by generating bounding boxes (i.e., not at pixel level).

Staniek [49] uses stereo vision cameras for the acquisition of road surface in the form of clouds of 3D points. The author emphasizes on solving stereo matching by using a Hopfield neural network, which is a special case of a recurrent artificial neural network. The author achieved 66% accuracy when evaluating matching pixels (for 50 image pairs) using a CoVar method [54] for evaluation..

The authors [50] have proposed a model to detect potholes based on YOLOv2 architecture. However, their reported architecture differs from our proposed model in *LM2*. Also the tested frames basically show not much more than potholes, while the real road scene is much complex. The CNN model proposed by the authors [51] to detect potholes has been trained on a CPU and experiments shows that CNN based model perform better than Conventional SVM based approach. However, the system is not able detect potholes under varying illumination conditions.

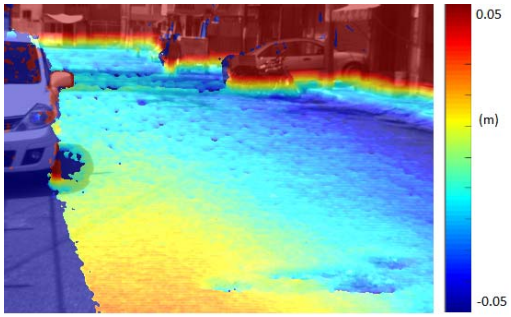

Fig. 4. Obtained elevation-difference map using a best-line fit in y-disparity space. In the applied color key, blue corresponds to points being 5 cm or more below the road surface, and red to points being 5 cm or more above the road surface).

The authors [52] have proposed a CNN based model mainly to classify a region on a road as pothole or non-pothole. The author has collected the dataset using smartphone camera mounted on the front windshield of the vehicle and the authors have used preprocessed cropped frames with ROI to train the proposed model. The authors [53]have developed CNN based model using thermal images to classify whether an image has pothole or not. The thermal images are recorded using a thermal camera.

III. PROPOSED AND TESTED METHODS

This section presents four different methods for pothole detection which are proposed by the authors. Comparative performance evaluations will be given later.

A. Single-Frame Stereo-Vision-Based Method - SV1

Using [34], we identify potholes by detecting areas within the road which are evaluated as being "below the road surface". Thus, for identifying a pothole, first we need to estimate the road manifold. There is a variety of techniques for modeling a road manifold; we applied the following two methods for obtaining a road manifold.

1) y-Disparity Line Fitting Model: A calculated dense disparity map D is used to compute the y-disparity map [68], [69] as follows:

$$V(y,d) = \text{card}\{x : 1 \le x \le N_{\text{cols}} \wedge D(x,y) = d\} \qquad (7)$$

where N_{cols} is the width of the input images in pixels, and $d \in [0, d_{\max})$ is a disparity bounded by the maximum value $d_{\max}$. $V(y,d)$ gives the number of pixels sharing the same disparity d in the y-th row of the disparity map.

Assuming that recorded images show "large" parts of the road surface, the lower envelope of data in the y-disparity map is approximated by a straight line $d = f(y)$ using a *random sample census* (RANSAC)-based approach. The estimated line $d = f(y)$ is applied to the disparity map D for identifying a road manifold. If $|D(x,y) - f(y)|$ is smaller than a threshold (say, 1) then the pixel at (x,y) is considered to show a 3D point on the road.

An elevation-difference map, obtained his way by the signed elevation difference to the road manifold, is shown in Fig. 4.

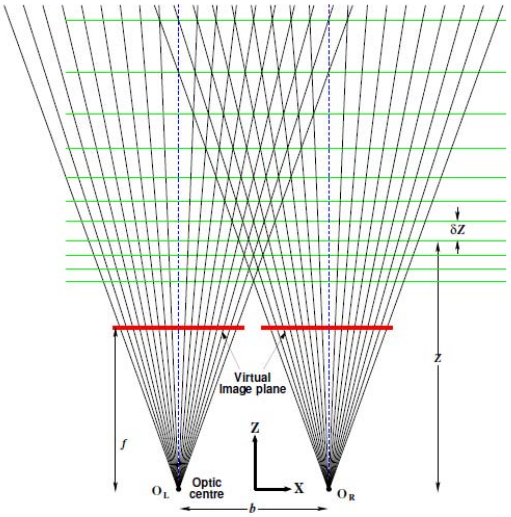

Fig. 5. Integral disparities define non-linearly distributed depth layers in 3D space. Courtesy of Waqar Khan [35].

The road surface is fairly planar in this scene close to the middle of the road, but bends downward to the right, towards the curb. Two potholes are visible in the lower right, not yet part of the general situation on the right of the road.

Note that disparities are not linearly related to depth; see, e.g., Fig. 5. Thus, this y-disparity-based approach can only be approximately correct in regions relatively close to the *ego-vehicle* (i.e. the vehicle the camera was operating in).

2) 3D Plane Fitting in Disparity Space: An (assumed) planar road manifold was approximated in [14] by directly considering disparities, not via a time-consuming 3D scene reconstruction way. A plane in 3D Euclidean space is represented by $a_0\,X + a_1\,Y + a_2\,Z + a_3 = 0$, where $a_0, \ldots, a_3 \in \mathbb{R}$ are the plane's coefficients. The planarity in 3D space is reserved in image-disparity space (x, y, d) due to perspective (i.e. pinhole) projection.

To define a plane uniquely, we use the normal-offset parametrization. Thus we use a unit 3-vector $\mathbf{n}$ and an offset parameter $\delta \in \mathbb{R}^+$, where

$$\delta = \frac{|a_3'|}{\sqrt{{a_0'}^2 + {a_1'}^2 + {a_2'}^2}} \tag{8}$$

and

$$\mathbf{n} = \frac{\delta}{a_3'} \cdot (a_0', a_1', a_2')^\top \tag{9}$$

A point $\mathbf{p} = (x, y, d)^\top$ is *up to δ on the plane* $(\mathbf{n}, \delta)$ if and only if it satisfies the following equation:

$$\mathbf{p}^\top \mathbf{n} - \delta = 0 \tag{10}$$

The *signed distance* of an off-plane point $\mathbf{p}$ is defined by

$$\varepsilon(\mathbf{p}; \mathbf{n}, \delta) = \mathbf{p}^\top \mathbf{n} - \delta \tag{11}$$

where the point is considered to be *above the plane* if $\varepsilon > 1$, with an up-vector defined by $\mathbf{n}$, and a point is *below the plane* if $\varepsilon < 1$.

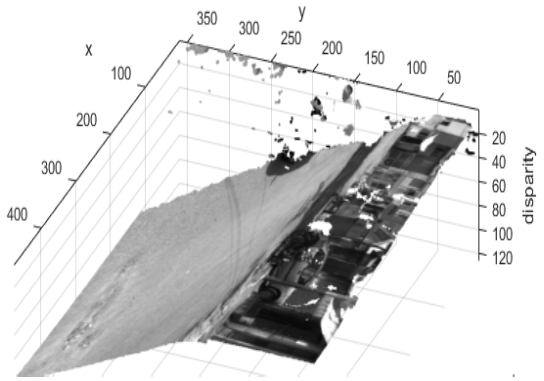

Fig. 6. Scene rendered in image-disparity space after dominant plane detection; the planarity of road pixels is assumed.

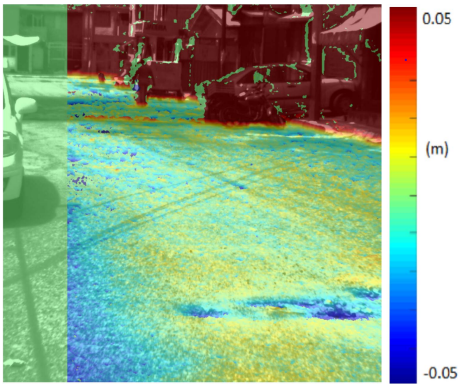

Fig. 7. Elevation-difference map after 3D plane fitting in disparity space; the used color key is again shown on the right.

Given a certain noise level, to consider a point to be an *inlier* with respect to a plane hypothesis $(\hat{\mathbf{n}}, \hat{\delta})$, we use a threshold $\varepsilon_{\max} \in \mathbb{R}^+$ if

$$|\varepsilon(\mathbf{p}; \hat{\mathbf{n}}, \hat{\delta})| \leq \varepsilon_{\max} \tag{12}$$

A RANSAC process, based on Eq. (12), is carried out to locate the *dominating plane* in the scene.

The process starts with a minimum set of points drawn from the population, and a hypothesis is obtained by finding the best fit plane of the randomly selected samples. To verify and support our hypothesis, first we find all the inliers from the population. If the hypothesis is supported by significantly many inliers, say 50% of the population, then it is considered a model candidate. Such process is repeated for a predefined number of iterations. In the end, the candidate model which is supported by highest number of inliers is considered as winner of the selection process. Following the described RANSAC process, a dominating plane is found in a robust manner. See Fig. 6 for an example.

This dominating plane can then again be used for finding pixels that are below the assumed planar road surface, analogously to the case of a y-disparity-based surface model. Following (11), we obtain an elevation-difference map. See Fig. 7.

After carefully comparing results, using either the y-disparity based road-surface approximation or the plane-approximation in disparity space for road manifold estimation, we decided for the latter one. Also compare

Figs. 4 and 7. Thus, *method SV1* is defined by plane-approximation in disparity space. To carry out further investigations, pixels more than one unit below, or $\varepsilon < 1$, are considered to be pothole candidates. We also use 8-adjacency connectedness analysis [70], [71] to remove regions larger than a reasonable size.

Method *SV1* considers individual *frames* (i.e. stereo pairs of images). The detected (planar) road manifold does not consider any bending of road surface (e.g., towards the curbs).

B. Multi-Frame Fusion-Based Method - SV2

The *digital elevation model* (DEM) approach in [6] provided a motivation for a multi-frame fusion in [15] for improved digital elevation models. For multi-frame integration, we first solve camera poses with respect to a reference frame using a *visual odometry* (VO) technique. After point clouds from different frames are aligned to the reference frame, we further transform them into a road-centered space, using a rigid transformation solved by means of *principal component analysis* (PCA). This section describes thus an improved strategy over the single-frame pothole detection method, described in the previous subsection.

1) Digital Elevation Model: It is common to represent a DEM by a regular grid of squares each labeled by a height difference to a zero-height plane. Figures. 4 and 7 showed elevation differences with respect to the perspective (i.e. pinhole) transform. Now we basically aim at a uniform height-difference representation across the recorded scene.

The construction of a DEM M is as follows. For each point $(X, Y, Z) \in \mathbb{R}$ within a *range of interest*, defined by the rectangle $X \in [X_{\min}, X_{\max}]$ and $Z \in [Z_{\min}, Z_{\max}]$ ahead of the ego-vehicle and an assumed height threshold $Y_{\max}$, an accumulating cell $(i, j) \in \mathbb{Z}^2$ is given by

$$i = \left[\frac{Z - Z_{\min}}{W}\right], \quad j = \left[\frac{X - X_{\min}}{W}\right] \tag{13}$$

where W is a chosen size of the used grid such that every cell spans an area of $W \times W$ in the XZ-plane, and $[a]$ is the nearest integer to real number a. The value $M(i, j)$ of cell (i, j) is decided by all the points assigned to cell (i, j). Here we build $M(i, j)$ by the averaged signed distance to the road plane, of all points in cell (i, j).

Now we use a PCA technique to find a rigid transformation that normalizes the point cloud in the given range of interest, such that the Z-axis aligns to the primal axis of the road, and the Y-axis is parallel to the normal vector $\mathbf{n}$ of the plane. After the transformation is applied to the point cloud, we define the transformed space as the *road-centered space* (RCS), on which the DEM is constructed.

So far this is all done for a single stereo frame only. Figure 8 shows a 3D visualization of a segment of a calculated DEM. There is a regular grid, bended by following the calculated values $M(i, j)$. The used color key for elevation differences is also shown at the bottom of the figure. Note that we could also show the DEM as a regular colored grid if deciding for a straight top-down view.

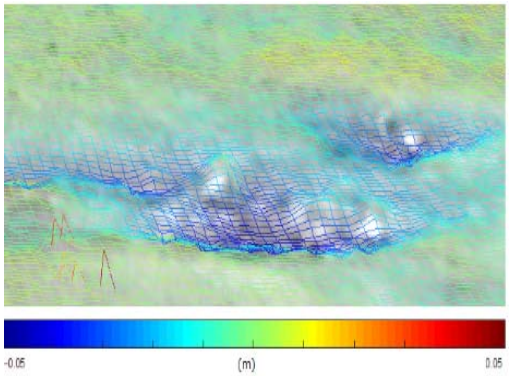

Fig. 8. Visualization of a segment of a DEM providing a close look at a pothole, using cell resolution of 1×1 cm^2.

2) Visual Odometry: To accumulate 3D data measured in different frames, their poses has to be recovered with respect to the chosen reference coordinate system. For example, consider that a reference coordinate system is taken for Frame t, and 3D data measured for Frame t up to Frame $t + m$ (e.g., $m = 5$) need to be mapped uniformly into the reference coordinate system of Frame t. To recover the self-motion of the camera from a video sequence, we implemented a 3-stage VO hybrid model as described in this subsection.

First stage. An efficient *perspective-from-n-point* (PnP) algorithm [72] is used, along with acquired 3D-to-2D mapping, to compute an initial pose $(\mathbf{R}, \mathbf{t})$ consisting of $\mathbf{R} \in SO(3)$, a rotation matrix, and $\mathbf{t} \in \mathbb{R}^3$, a translation vector.

Second stage. During this stage, we adopt the SURF feature detector and descriptor extractor [73] and camera intrinsics to derive an *essential matrix*, to calculate Sampson distance for each pair of matched key points (see [74] for details). Subject to the filtered correspondences, the reprojection error is minimized using the *Levenberg-Marquardt* (LM) method [75] in the sum-of-squares form:

$$\varphi_{\text{RPE}}(\mathbf{R}, \mathbf{t}) = \sum_{i}^{\mathcal{P}} \left\| \pi \left(\mathbf{R} \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} + \mathbf{t} \right) - \begin{bmatrix} x_i' \\ y_i' \end{bmatrix} \right\|^2 \tag{14}$$

where $\mathcal{P}$ denotes a set of sparse 3D-to-2D correspondences, and $\pi : \mathbb{R}^3 \to \mathbb{R}^2$ symbolizes the pinhole projection function.

Third stage. During this stage, we warp the left image I_L of the last stereo frame (i.e. Frame $t + m$ in our example above) to the current frame using a pose hypothesis being tuned, and iteratively reducing the sum-of-squares difference in intensities. We use an LM algorithm to approach a local minima of (15), starting with the pose previously minimized subject to (14). The objective function here is defined as follows:

$$\varphi_{\text{INT}}(\mathbf{R}, \mathbf{t}) = \sum_{i}^{\mathcal{Q}} \left[I_L' \left(\pi \left[\mathbf{R} \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} + \mathbf{t} \right] \right) - I_L(x_i, y_i) \right]^2 \tag{15}$$

where $\mathcal{Q}$ is a set of pixels with valid depth data.

3) Weight Assignment and Multi-Frame Fusion: We adopt weighted averaging and a weight is derived, for each data

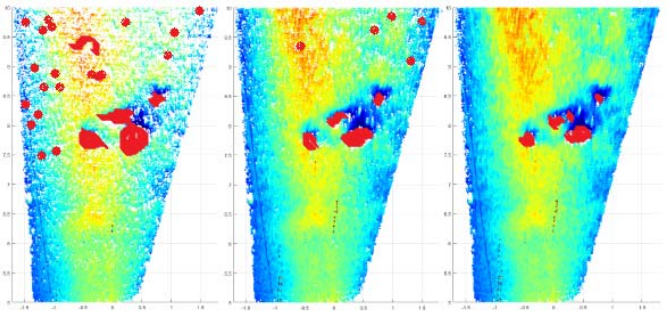

Fig. 9. From left to right: Road surfaces from 20 accumulated frames before and after hole filling, identified local minima (as highlighted by red marks).

point, by evaluating its disparity. Given the observed left image I_L, the block-wise correlation is defined as

$$C(x,y) = \sum_{p \in A(x,y)} \frac{\left(I_L(p) - \mu_{xy}\right)\left(I'_L(p) - \mu'_{xy}\right)}{\sigma_{xy}\sigma'_{xy}} \qquad (16)$$

where $A(x,y)$ denotes neighbors centering at pixel (x,y), μ_{xy} and σ_{xy} are local mean and standard deviation, respectively, calculated from $A(x,y)$ in image I_L, and μ'_{xy} and σ'_{xy} are those calculated from the reconstructed image I'_L.

For a good disparity estimate in (x,y), the correlation $C(x,y)$ will be close to 1, while an inaccurate estimate will lead to a low coefficient, as low as -1. We use a normalized indicator $W(x,y) = (C(x,y)+1)/2$ to weight each point during the accumulation process.

In Fig. 9, accumulations over 5, 10, and 20 frames of a tested sequence are rendered. If depth data from more frames are integrated, the resulting DEM becomes denser and presents less missing cells.

Method SV2 is defined by the described accumulation approach. Potholes are detected in accumulated DEMs as in method *SV1* (e.g. with connected-component analysis).

Comparing *SV2* to the single frame approach *SV1* described in Subsection III-A.2, the multi-frame DEM approach not only models the road manifold in a more reliable way, but also provides more accurate geometric measures such as depth and size of each pothole. Morphological opening operations in *SV2* follow, as for *SV1*, ideas published by Z. Zhang [34] with some slight modifications. We modeled the road manifold for *SV2* (within a local context) by using a second order polynomial curve as profile, to consider also twisting or bending of the surface of the road.

C. Using Transfer Learning With Mask R-CNN - LM1

Inspired by the results of CrackNet [67], we applied transfer learning [77] using Mask R-CNN [78]. It predicts a soft mask to delineate the boundary of each instance at pixel level.

The origin of Mask R-CNN is *region-based convolutional neural network* (R-CNN) [79], published in 2014. The R-CNN incrementally improved into Fast R-CNN [81] and then Faster R-CNN [82].

The backbone of Mask R-CNN implementation uses ResNet101 [83] and FPN [84]. A ResNet is a standard feature extractor which detects low-level features at early layers, and high-level features at later layers. The network accepts an image of $1,024 \times 1,024$. The image is padded with zeroes if it is not given as per the assumed aspect ratio. FPN is another improved feature extractor, a second pyramid that allows features at every level to have access to both lower- and higher-level features.

Mask R-CNN is composed of a two-step framework. The first step scans the whole image for generating proposals. The second step classifies the proposals, generates bounding boxes, and also masks of an object.

The main modules of Mask R-CNN are as follows:

1) Region Proposal Network: RPN is a lightweight neural network that scans over the backbone's feature map, using a sliding window to generate anchors that are typically boxes distributed over the image area. The sliding window operation is handled by its convolutional nature, and this is very fast on a GPU. The output of the RPN is an anchor class and a bounding-box refinement.

2) Refined Bounding Boxes: To precisely map bounding boxes to the regions of an image, Mask R-CNN also improves the RoIAllign layer of the network for pixel-level segmentation. It removes the harsh quantization of the RoIPool layer to properly encapsulate the extracted features with the input.

3) Instance Masks: The mask branch is a CNN that accepts positive regions as input generated by the classifier, and predicts a low resolution 28×28 soft mask for it. A soft mask differs from a binary mask as these are represented by float numbers and hold more details.

The main working steps of Mask R-CNN are as follows:

- Retraining an RPN end-to-end for a region proposal network, initialized by a pre-trained CNN image classifier. $\texttt{IoU} > 0.7$ and < 0.3 define positive or negative samples, respectively. To formally apply Intersection over union IoU, a ground truth box G and a predicted bounding box P are used, and define
$$\texttt{IoU} = \frac{|G \cap P|}{|G \cup P|} \qquad (17)$$
where $G \cap P$ is the intersection, $G \cup P$ the union, and $|.|$ the of the resulting sets.
- A small $n \times n$ window slides over the convolved feature map of the entire image.
- Anchor is produced to predict the multiple regions, at each sliding position.
- Train an object detection model by using the proposals obtained by RPN.
- Fine tune the layer of Mask R-CNN, according to the object class name.

For transfer learning, we used weights trained on the *COCO* [85] dataset and adapted the weights to identify potholes. We used 247 images for training dataset, where 50 of the CCSAD's Urban Sequence 1, 100 of DLR, 48 Japan and 49 ofSunny dataset collectively and validation dataset comprised of 50 frames are also selected using same datasets.

For test dataset, we used 50 images from CCSAD's sequences 2 and very challenging PNW dataset. We also used data augmentation techniques in order to compensate for less data. For *LM1* we use left-right flips for augmentation.

TABLE V
EVALUATION MEASURES (IN %) FOR VALIDATION DATASET USING TECHNIQUE LM1 AS SHOWN IN FIG. 12

Dataset	Frame	Precision	Recall
Japan	20170912135214	72.8	65.7
DLR	472000	67.7	88.3
DLR	572000	100.0	100.0
DLR	749000	100.0	92.2
DLR	449000	91.0	36.1
Japan	20170906135035	76.8	86.9
Japan	20170906135037	96.4	72.8
Sunny	G0010116	73.9	26.6
Sunny	G0010118	100.0	100.0
Sunny	G0011873	78.5	50.0

We excluded 6 frames from CCSAD's Urban Streets sequences 1 for comparison purpose with *SV2*. We started with learning rate of 0.01 and it resulted in very large update to weights. As we used keras with TensorFlow, the optimizers are implemented differently. Therefore, we set learning rate as 0.001 and did not lower it further. We train the network using stochastic gradient descent and a learning momentum of 0.9. We used a batch size of 2 and for 30 epochs it took 14 hours on a GeForce GTX 1080 GPU.

As we used images from four different datasets, the image dimensions were different. To keep an aspect ratio of uniform size $1,024 \times 1,024$, zero padding is added to the top and bottom of an image.

We have two classes in our dataset, one for "background" and one for "pothole". Transfer learning with Mask R-CNN is a two-stage framework as follows:

Stage 1: Classification and bounding box refinement. During the first stage, the whole training image is scanned to generate anchor proposals by fine tuning RPN from end-to-end. RPN is a lightweight neural network that scans over the backbone's feature map, using a sliding window to generate anchors. Anchors are typically bounding boxes in the image to predict multiple regions while a small $n \times n$ window slides over the convolved feature map of the entire training image. As the sliding-window operation is convolutional in nature, so it is handled fast on a GPU. This stage generates a maximum of 256 anchors per image and bounding-box refinements. This stage outputs a grid of anchors at different scales.

Here IoU > 0.7 define positive samples, and IoU < 0.3 define negative samples, respectively.

The bounding box refinement step accepts a refined grid of anchors from the RPN and classifies the anchors precisely. It maps anchor bounding boxes into final boxes. Mask-RCNN refines the ROIAllign layer by removing a harsh quantization of the RoIPool layer, to properly encapsulate the extracted features with the input.

Stage 2: Mask generation.

The mask branch is a CNN that accepts positive regions as input, generated by the classifier during Stage 2, and generates a low resolution 28×28 soft mask for it. We fine-tune the layer of Mask-RCNN, according to our object class "pothole". Some frames from validation dataset are shown in Fig. 12 and evaluation measures are listed in Table V. All the images from validation datasets show potholes being identified with high level of accuracy.

D. Using Transfer Learning With YOLOv2 - LM2

For real time detection of potholes, we used transfer learning using another object detector- *You only look once* (YOLOv2). YOLOv2 [86] based on regression algorithm, predicts classes and bounding boxes for the whole image. A single CNN is used for both classification and localization of an object in YOLO.

Our YOLOv2 experiments is based on 22 convolutional layers and 5 maxpool layers.

The input image gets divided into a $c \times c$ grid cell. The purpose of that grid cell is to find that object, whose center falls into particular grid cell. In YOLOv2 the input image size is 416×416 with five maxpool layers. So, grid cell size is $(416 \times 416)/pow(2,5) = 13$. These grid cells produces N bounding boxes along with their confidence scores. Next step is to perform non-max suppression, which is a process of removing bounding boxes with low object probability and highest shared area.

The bounding box consists of 5 numeric predictions: confidence, x, y, width, height. where confidence score represents *Intersection over Union* IoU between predicted and ground truth box. x, y are coordinates center of box relative to grid cell, width and height are relative to input. During testing the confidence score represents how likely and accurate the bounding box has the object and the bounding box with higher IoU is selected. The loss function in YOLO is mainly comprises of: classification, localization and confidence loss.

We started with setting the input image subdivision value as 8, but due to a resulting high memory requirement we changed it to 32. We used 64 images per batch. Initially, the learning rate was set to 0.01. However, after $1,000$ iterations the average loss kept on increasing.

Therefore, we used 0.0001 for learning rate. We trained the *LM2* network using TESLA K80 GPU. For around $8,000$ iterations it took approximately $6-7$ hours and checked the *mean average precision* mAP [87] value for different iterations at $5,400$, $6,400$, $7,400$. The mAP value of $5,400$ iterations was higher than other iteration weights. Hence, using Early Stopping Point, the model is selected after $5,400$ iterations. After $5,400$ iterations the average loss was same, hence we stopped training at $5,400$ iterations. The dataset for training and testing was same for *LM1* and *LM1*. In *LM2* we use random rotation techniques for augmentation. However, the annotation format is different in case of *LM2*, which is bounding box not mask. Table VIII shows IoU measure values for a threshold of 0.5 for some of the selected frames of PNW dataset.[1] Some of the frames are shown in Fig. 10.

We also tested 50 randomly selected PNW frames at different IoU values 0.5, 0.6 and 0.7 and the obtained average precision values are 60%, 52% and 45%.

IV. DATASETS

Authors of [64] state that "there is no uniform road damage dataset available openly, leading to the absence of a benchmark for road damage detection". Existing datasets on websites of

[1] A short video sequence of results (using *LM2*) on the extensive PNW test dataset can be seen here: https://vimeo.com/337886918. This video is 29 fps.

Fig. 10. Detected "potholes" using *LM2* method, shown in two columns with original image on the left and predicted results on the right.

Fig. 11. Top left - *CCSAD*, Top right - *DLR*, middle left - Japan, middle right- Sunny, bottom - PNW.

the KITTI [47] or *.eisats.*. [88] projects, or of Middlebury College [89] have been recorded in countries where pothole on roads typically occurs rarely, and they are not benchmarks for road pothole. Under these circumstances, we decided for the use of the following data (see Fig. 11 for examples):

1. *CCSAD*. Hayet *et al.* [90] has introduced a dataset of *challenging sequences for autonomous driving* (CCSAD) that is exceptionally utilitarian to execute strategies for detection of road pothole in Mexico. The CCSAD dataset has been split into four parts `Colonial Town Streets`, `Urban Streets`, `Avenues and Small Roads`, and `Tunnel Network` which accounts for 500 GB of data that incorporates calibrated and rectified pairs of stereo images, videos and meta-data.

The *CCSAD* `Urban Streets` dataset is an extensive collection of road potholes. The data has been acquired at 20 fps using two Basler Scout scA1300-32fm firewire greyscale cameras mounted on the roof of the car. The image resolution of `CCSAD` is 1096×822.

2. `DLR`. This dataset has been recorded while using the *integrated positioning system* (IPS) [92], developed by the German Aerospace Centre (DLR), installed on a car. The collected dataset accounts 288 GB for with image dimension as $1360 * 1024$.

3. `Japan`. This dataset comprises of $163,664$ road images of dimension 600×600 collected in seven different cities of Japan [64]. The dataset contains $9,053$ damaged-road images and $15,435$ instances of damaged road surfaces such as (mainly) cracks and (rarely) potholes. Images are captured at an interval of one second under different weather and lighting conditions.[2]

4. `Sunny`. Authors of [93] provided a dataset of $48,913$ images of size $3,680 \times 2,760$ recorded using a GoPro camera, mounted inside a car on its windscreen. The camera was set to a 0.5 second time lapse mode and car was moving at an average speed of 40 km/h while scanning the road surface. The total data available is 2.70 GB.

5. `PNW`. PNW is an extensive video recorded on the Pacific Northwest highway [94]. It shows the highway with patches of snow and water. We used $19,784$ extracted frames of dimension 1280×720 from this video.

V. Experimental Evaluations

Using *SV1, SV2, and LM1* methods, we detected potholes at pixel level segmentation and obtained results are very

[2]In late 2018, these data have been used in a competition, see bdc2018.mycityreport.net/overview/.

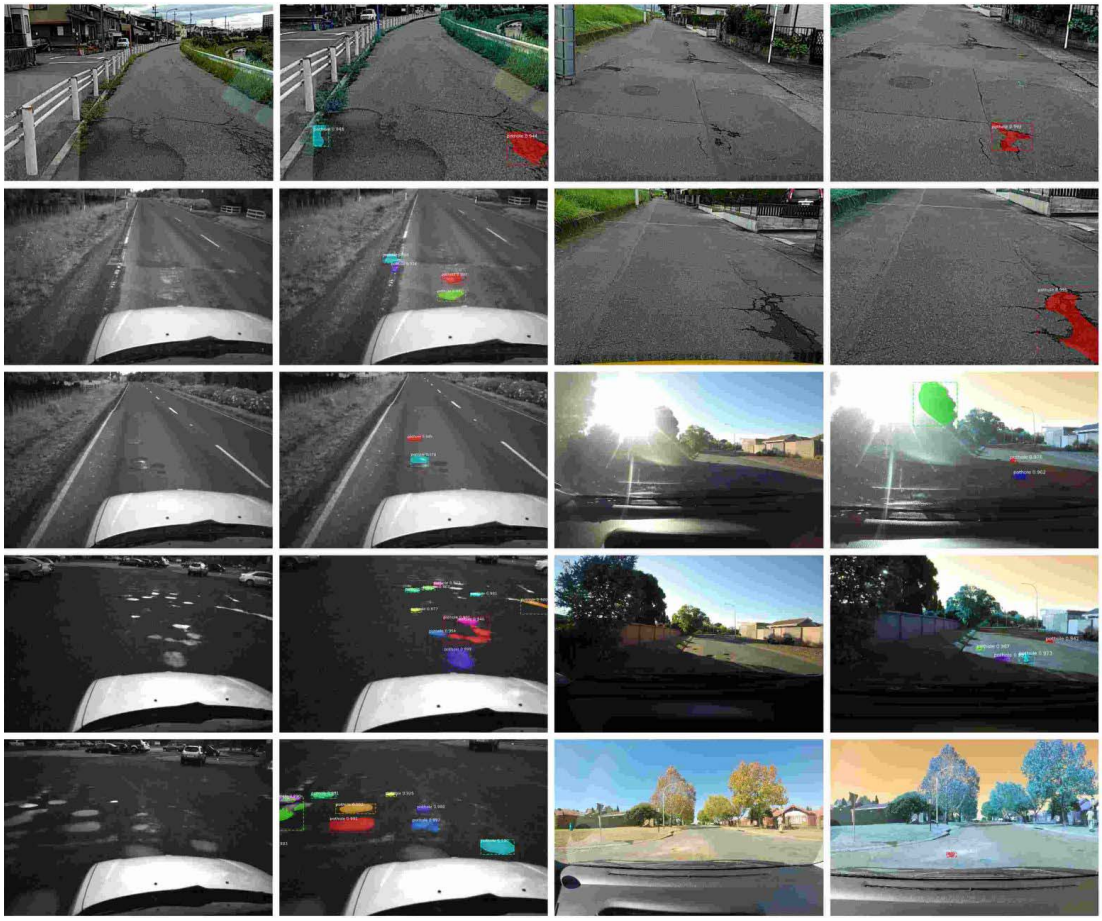

Fig. 12. Detected "potholes" from validation dataset using LM1 method, shown in two columns with original image on the left and predicted results on the right. *Top to bottom, left to right*: Ten frames in order as listed in Table V.

Fig. 13. Examples of false detections from test PNW dataset using *LM1*. The detection of pothole including false positives varies from 3070 to 3076 frames shown in top two rows.

promising. Figure 12 shows that in validation dataset pothole instances are correctly identified while a false positive has been detected in the third image (from the bottom) - as a pothole is of arbitrary shape, under bright sunshine a tree is miss-classified as a pothole in this case (this could be excluded by identifying a ground manifold first).

Fig. 14. Pothole marked in red color presents a very complex situation as it is filled with water and shadow of a tree.

TABLE VI
COMPARATIVE EVALUATION OF THE PROPOSED *LM1*, *SV2* AND *SV1* IN %

Frame	Precision			Recall		
	LM1	SV2	SV1	LM1	SV2	SV1
0078	**96.2**	65.2	67.4	**92.2**	53.2	46.6
0093	**93.1**	64.0	57.3	**93.7**	61.4	69.2
0278	**84.6**	63.1	78.2	**94.1**	52.9	41.0
0304	80.7	**92.5**	76.5	**84.4**	57.6	72.8
0547	**89.0**	81.5	37.1	**95.6**	34.7	24.0
0935	**95.4**	69.7	24.2	**97.1**	49.1	45.1
Means	**89.8**	67.4	45.8	**92.8**	51.2	45.8

More examples of false detections using *LM1* are shown in Fig. 13 where frames in top two rows are from continuous frame number as 3070 to 3076. Frames in this range have only one pothole yet its been detected accurately in frame number 3073 (second row first frame). It is interesting to see that false detection in this frame range is because pothole is filled with water and tree shadow (see Fig. 14).

In order to measure accuracy of our models, we mainly calculate common classification measures `Precision` and `Recall` on a per-pixel basis. Precision measures the correctness among all positive pothole instances, recall measures how many positive pothole instances are successfully printed among all positive pothole instances. – For method *LM2*, we use `IoU` as an evaluation measure.

We comparatively tested the *SV1, SV2*, and the proposed *LM1* technique. Results of *LM1*, are great improvement compared to the same example-frames as shown in Fig. 15 when using the *SV1* or *SV2* approach.

Table VI lists results for a few frames of the tested `CCSAD Urbban Sequence 1`, comparing pixel-wise detected potholes with the manually labeled ground truth. The table shows that there is a case where the *SV2* method provides a *slightly* better result, but it also demonstrates the general observation that the *LM1* method outperforms the *SV1* method in a majority of cases, and typically by providing *much* better results.

Table VII shows precision and recall values of testing dataset `PNW` for some of the frames, see Fig. 16. This method miss classifies a pothole in `PNW` frame 720. The reason is that network identified a region as pothole which is a darker region on the side of pothole. However, *LM1* method has great potential to identify potholes whether they are dry, or filled with water or snow. The overall `precision` and `recall` for randomly chosen 50 frames from our testing dataset dataset is 88% and 84% respectively.

Fig. 15. Left column shows *SV1* and *SV2* results, marked in red and green, respectively, with blue for ground truth. The right column shows results for *LM1*.

TABLE VII
EVALUATION MEASURES FOR TEST DATASET USING *LM1*; IN %

Dataset	Frame	Precision	Recall
PNW	266	100.0	100.0
PNW	720	41.2	76.6
PNW	1710	84.7	81.7
PNW	1756	82.9	84.3
PNW	1871	100.0	100.0
PNW	2159	100.0	70.7
PNW	2832	100.0	67.1
PNW	18343	100.0	100.0

Table VIII lists `IoU` values for some of the chosen frames, see Fig. 10 for `PNW` testing dataset using *LM2* method. The mean is 69% because to annotate pothole which is always of irregular shape, using bounding box is very difficult. We also considered about pothole instance count matching as an

Fig. 16. Detected "potholes" using *LM1* method, shown in two columns with original image on the left and predicted results on the right.

TABLE VIII

EVALUATION MEASURES (IN %) FOR *LM2* RESULTS SHOWN IN FIG. 12; NOTE THAT BB1 REPRESENTS BOUNDING BOX 1, AND BB2 BOUNDING BOX 2 FOR DIFFERENT INSTANCES OF POTHOLES

Frame	IoU	Frame	IoU
122	66	8464	68
4667	66	8527(bb1)	76
4685	79	8527(bb2)	79
4691	90	8540	55
7044(bb1)	76	8664	67
7044(bb2)	75	8874(bb1)	45
7335	60	8874(bb2)	55
7337	60	10064	83
7476	52	10595	90
8133	51	12601	89
Averaged			69

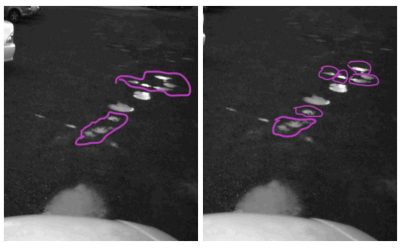

Fig. 17. The potholes marked in purple colors can be perceived as one big pothole or can be counted separately(two potholes in left image, six potholes in right image)[other potholes are not marked here, but considered in experiments].

evaluation measure. However, as shown in Fig. 17, more than one pothole are adjacent to each other can be identified as one big pothole or multiple small potholes. In conclusion, we did not include pothole counts as an evaluation measure because it depends on individual counting.

Using the *LM2* method, the developed model is applicable for real-time scenarios. In Table VIII, IoU is greater than 50%, thus we may say that results are promising. The PNW test frames get divided into the same number of grids as selected during the training period, i.e. 13×13. The model can predict multiple bounding boxes in each grid, so we keep the one with the highest IoU value. This leads to an enforcement of spatial diversity in making predictions.

VI. CONCLUSION

The gravity of pothole related accidents can be understood by increased numbers of accidents around the world due to potholes. In this research, four different techniques are proposed and tested against each other. Each technique has its own benefits and can provide different pathways to a number of applications. The *LM1* model can identify a pothole under challenging weather conditions with good precision and recall whereas the *LM2* model is capable of real-time pothole identification. The *SV2* approach can identify potholes and road manifolds with very high accuracy when used with stereo-vision cameras. The *SV2* approach can also be used to track a pothole from one frame to another, and is relatively easy to implement.

The findings that we have presented here suggest that it is very difficult to define the irregular shape of a pothole which further makes it difficult to annotate ground truth. This, in turn, causes a complex process of matching results with ground truth. To date, there is no platform or benchmark available for pothole identification. As a result of conducting this research, we also put forward six datasets specifically for pothole identification, and discussed applications of two different areas of research such as computer vision and deep

learning. It would be fruitful to pursue further research in order to combining the output of *LM1* for annotating pothole data and to use it to train more *LM2*-type models in order to increase detection accuracy for real-time purposes.

ACKNOWLEDGMENT

The authors acknowledge fruitful discussions with Hsiang-Jen Chien, Auckland, New Zealand, which have been very helpful at various steps of the reported work.

REFERENCES

[1] A. Heaton, "Potholes and more potholes: Is it just us," Tech. Rep., Mar. 2018. [Online]. Available: https://medium.com
[2] (2018). *Pothole Facts*. [Online]. Available: www.pothole.info/the-facts
[3] N. Dwivedi, "The pothole proposition," Tech. Rep., Aug. 2018. [Online]. Available: https://medium.com
[4] (2018). *Christchurch Report*. [Online]. Available: www.stuff.co.nz/the-press/news/100847641/christchurch-the-pothole-capital-of-new-zealand/
[5] H. Kong, J.-Y. Audibert, and J. Ponce, "General road detection from a single image," *Image Process.*, vol. 19, no. 8, pp. 2220–2221, Aug. 2010.
[6] X. Ai, Y. Gao, J. G. Rarity, and N. Dahnoun, "Obstacle detection using U-disparity on quadratic road surfaces," in *Proc. Int. Conf. Intell. Transp. Syst.*, Oct. 2013, pp. 1352–1357.
[7] F. Oniga, S. Nedevschi, M. M. Meinecke, and T. B. To, "Road surface and obstacle detection based on elevation maps from dense stereo," in *Proc. Int. Conf. Intell. Transp. Syst.*, Oct. 2007, pp. 859–865.
[8] C. Koch and I. Brilakis, "Pothole detection in asphalt pavement images," *Adv. Eng. Inform.*, vol. 25, no. 3, pp. 507–515, 2011.
[9] (2018). *Driverless Car Market Watch*. [Online]. Available: http://www.driverless-future.com/?page_id=384
[10] A. Tedeschi and F. Benedetto, "A real-time automatic pavement crack and pothole recognition system for mobile Android-based devices," *Adv. Eng. Inform.*, vol. 32, pp. 11–25, Apr. 2017.
[11] B.-H. Lin and S.-F. Tseng, "A predictive analysis of citizen hotlines 1999 and traffic accidents: A case study of taoyuan city," in *Proc. Int. Conf. Big Data Smart Comput.*, Feb. 2017, pp. 374–376.
[12] D. Santani *et al.*, "Communisense: Crowdsourcing road hazards in nairobi" in *Proc. Int. Conf. Hum.-Comput. Interact. Mobile Devices Services*, Aug. 2015, pp. 445–456.
[13] D. O'Carroll, "For the love of pizza, Domino's is now fixing potholes in roads," Wellington, New Zealand, Tech. Rep., Jun. 2018. [Online]. Available: https://stuff.co.nz
[14] A. Dhiman, H.-J. Chien, and R. Klette, "Road surface distress detection in disparity space," in *Proc. Int. Conf. Image Vis. Comput. New Zealand*, Dec. 2017, pp. 1–6.
[15] A. Dhiman, H.-J. Chien, and R. Klette, "A multi-frame stereo vision-based road profiling technique for distress analysis," in *Proc. ISPAN*, Oct. 2018, pp. 7–14.
[16] A. Dhiman, S. Sharma, and R. Klette, *Identification of Road Potholes*. Stratford, U.K.: MIND, 2019.
[17] A. Mednis, G. Stardins, R. Zviedris, G. Kanonirs, and L. Selavo, "Real time pothole detection using Android smartphones with accelerometers," in *Proc. Int. Conf. Distrib. Comput. Sensor Syst. Workshops*, Jun. 2011, pp. 1–6.
[18] M. Ghadge, D. Pandey, and D. Kalbande, "Machine learning approach for predicting bumps on road," in *Proc. Int. Conf. Appl. Theor. Comput. Commun. Technol.*, Oct. 2015, pp. 481–485.
[19] F. Seraj, B. J. van der Zwaag, A. Dilo, T. Luarasi, and P. Havinga, "RoADS: A road pavement monitoring system for anomaly detection using smart phones," in *Proc. Int. Workshop Modeling Social Media*, Jan. 2014, pp. 128–146.
[20] J. Ren and D. Liu, "PADS: A reliable pothole detection system using machine learning," in *Proc. Int. Conf. Smart Comput. Commun.*, Jan. 2016, pp. 327–338.
[21] K. Georgieva, C. Koch, and M. König, "Wavelet transform on multi-GPU for real-time pavement distress detection," in *Proc. Comput. Civil Eng.*, May 2015, pp. 99–106.
[22] K. Doycheva, C. Koch, and M. König, "Implementing textural features on GPUs for improved real-time pavement distress detection," *Real-Time Image Process.*, vol. 33, pp. 1–12, Sep. 2016.
[23] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *Syst. Man*, vol. SMC-3, no. 6, pp. 610–621, Nov. 1973.
[24] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd ed. San Mateo, CA, USA: Morgan Kaufmann, 2016.
[25] M. V. Thekkethala and S. Reshma, "Pothole detection and volume estimation, using stereoscopic cameras," in *Proc. Int. Conf. Mixed Design Integr. Circuits Syst.*, 2016, pp. 47–51.
[26] S.-K. Ryu, T. Kim, and Y.-R. Kim, "Image-based pothole detection system for ITS service and road management system," *Math. Problems Eng.*, vol. 2015, 2015, Art. no. 968361.
[27] L. Powell and K. G. Satheeshkumar, "Automated road distress detection," in *Proc. Int. Conf. Emerg. Technol. Trends*, 2016, pp. 1–6.
[28] A. Rasheed, K. Kamal, T. Zafar, S. Mathavan, and M. Rahman, "Stabilization of 3D pavement images for pothole metrology using the Kalman filter," in *Proc. Int. Conf. Intell. Transp. Syst.*, 2015, pp. 2671–2676.
[29] H. Hirschmüller, "Accurate and efficient stereo processing by semi-global matching and mutual information," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 807–814.
[30] Q. Li, M. Yao, X. Yao, and B. Xu, "A real-time 3D scanning system for pavement distortion inspection," *Meas. Sci. Technol.*, vol. 21, no. 8, pp. 015702-1–015702-8, 2010.
[31] X. Yu and E. Salari, "Pavement pothole detection and severity measurement using laser imaging," in *Proc. Int. Conf. Electro/Inf. Technol.*, 2011, pp. 1–5.
[32] K. K. Vupparaboina, R. R. Tamboli, P. M. Shenu, and S. Jana, "Laser-based detection and depth estimation of dry and water-filled potholes: A geometric approach," in *Proc. Nat. Conf. Commun.*, 2015, pp. 1–6.
[33] R. Klette, *Concise Computer Vision: An Introduction Into Theory and Algorithms*. London, U.K.: Springer, 2014.
[34] Z. Zhang, X. Ai, C. K. Chan, and N. Dahnoun, "An efficient algorithm for pothole detection using stereo vision," in *Proc. Int. Conf. Acoust., Speech Signal Process.*, 2014, pp. 564–568.
[35] W. Khan, "Accuracy of stereo-based object tracking in a driver assistance context," Ph.D. dissertation, Dept. Comput. Sci., Auckland Univ., Auckland, New Zealand, 2013.
[36] H. Youquan, W. Jian, Q. Hanxing, Z. Wei, and X. Jianfang, "A research of pavement potholes detection based on three-dimensional projection transformation," in *Proc. Int. Conf. Image Signal Process.*, 2011, pp. 1805–1808.
[37] C. Zhang and A. Elaksher, "An unmanned aerial vehicle-based imaging system for 3D measurement of unpaved road surface distresses," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 27, no. 2, pp. 118–129, 2012.
[38] Y.-W. Hsu, J. W. Perng, and Z.-H. Wu, "Design and implementation of an intelligent road detection system with multisensor integration," in *Proc. Int. Conf. Mach. Learn. Cybern.*, 2016, pp. 219–225.
[39] T. Naidoo, D. Joubert, T. Chiwewe, A. Tyatyantsi, B. Rancati, and A. Mbizeni, "Visual surveying platform for the automated detection of road surface distresses," in *Proc. Int. Conf. Sensors MEMS Electro-Optic Syst.*, 2014, Art. no. 92570D.
[40] F. Orhan and P. E. Eren, "Road hazard detection and sharing with multimodal sensor analysis on smartphones," in *Proc. Int. Conf. Next Gener. Mobile Apps Services Technol.*, 2013, pp. 56–61.
[41] T. Garbowski and T. Gajewski, "Semi-automatic inspection tool of pavement condition from three-dimensional profile scans," *Intell. Transp. Syst.*, vol. 172, pp. 310–318, Jan. 2017.
[42] *FEMat Project*. Accessed: May 20, 2019. [Online]. Available: www.fematproject.pl/index.html
[43] T. Shen, G. Schamp, and M. Haddad, "Stereo vision based road surface preview," in *Proc. Int. Conf. Intell. Transp. Syst.*, 2014, pp. 1843–1849.
[44] V. A. Bashkar and G. T. Manohar, "Surface pothole depth estimation using stereo mode of image processing," *Advance Res. Eng. Technol.*, vol. 4, pp. 1169–1177, Jan. 2016.
[45] Y.-H. Tseng, S.-C. Kanga, J.-R. Changb, and C.-H. Leea, "Strategies for autonomous robots to inspect pavement distresses," *Autom. Construct.*, vol. 20, no. 8, pp. 1156–1172, 2011.
[46] Z. Ying, G. Li, X. Zang, R. Wang, and W. Wang, "A novel shadow-free feature extractor for real-time road detection," in *Proc. Int. Conf. Pervas. Ubiquitous Comput.*, 2016, pp. 611–615.
[47] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.
[48] T. Veit, J.-P. Tarel, P. Nicolle, and P. Charbonnier, "Evaluation of road marking feature extraction," in *Proc. Int. Conf. ITSC*, Beijing, China, 2008, pp. 174–181.

[49] M. Staniek, "Neural networks in stereo vision evaluation of road pavement condition," in *Proc. Int. Symp. Non-Destructive Test. Civil Eng.*, 2015, pp. 15–17.
[50] L. K. Suong and K. Jangwoo, "Detection of potholes using a deep convolutional neural network," *Universal Comput. Sci.*, vol. 24, no. 9, pp. 1244–1257, 2018.
[51] V. Pereira, S. Tamura, S. Hayamizu, and H. Fukai, "A deep learning-based approach for road pothole detection in timor leste," in *Proc. Int. Conf. Service Oper. Logistics, Informat.*, 2018, pp. 279–284.
[52] K. E. An, S. W. Lee, S.-K. Ryu, and D. Seo, "Detecting a pothole using deep convolutional neural network models for an adaptive shock observing in a vehicle driving," in *Proc. Int. Conf. Consumer Electron.*, 2018, pp. 1–2.
[53] Y. Bhatia, R. Rai, V. Gupta, N. Aggarwal, and A. Akula, "Convolutional neural networks based potholes detection using thermal imaging," *King Saud Univ., Comput. Inf. Sci.*, to be published. doi: 10.1016/j.jksuci.2019.02.004.
[54] B. Cyganek and J. P. Siebert, *An Introduction to 3D Computer Vision Techniques and Algorithms*. Hoboken, NJ, USA: Wiley, 2011.
[55] A. Mikhailiuk and N. Dahnoun, "Real-time pothole detection on TMS320C6678 DSP," in *Proc. Int. Conf. Imag. Syst. Techn.*, 2016, pp. 123–128.
[56] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2017.
[57] G. Lin, A. Milan, C. Shen, and I. D. Reid, "RefineNet: Multi-path refinement networks for high-resolution semantic segmentation," *CoRR*, vol. abs/1611.06612, 2016.
[58] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," *CoRR*, vol. abs/1612.01105, 2016.
[59] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large kernel matters—Improve semantic segmentation by global convolutional network," *CoRR*, vol. abs/1703.02719, 2017.
[60] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. CVPR*, 2015, 3431–3440.
[61] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, 2017.
[62] H. Song, K. Baek, and Y. Byun, "Pothole detection using machine learning," *Adv. Sci. Technol. Lett.*, vol. 150, pp. 151–155, Feb. 2018.
[63] C. Szegedy, V. Vanhoucke, S. Ioffe, J. SHlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. CVPR*, 2016, pp. 2818–2826.
[64] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, and H. Omata, "Road damage detection using deep neural networks with images captured through a smartphone," 2018, *arXiv:1801.09454*. [Online]. Available: https://arxiv.org/abs/1801.09454
[65] J. Huang *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors," in *Proc. CVPR*, 2017, pp. 7310–7311.
[66] A. G. Howard *et al.*, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*. [Online]. Available: https://arxiv.org/abs/1704.04861
[67] A. Zhang *et al.*, "Automated pixel-level pavement crack detection on 3D asphalt surfaces using a deep-learning network," *J. Comput.-Aided Civil Infrastruct. Eng.*, vol. 32, no. 10, pp. 805–819, 2017.
[68] R. Labayrade, D. Aubert, and J.-P. Tarel, "Real time obstacle detection in stereovision on non flat road geometry through 'v-disparity' representation," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2002, pp. 646–651.
[69] N. H. Saleem, H.-J. Chien, M. Rezaei, and R. Klette, "Improved stixel estimation based on transitivity analysis in disparity space," in *Proc. Int. Conf. Comput. Anal. Images Patterns*, 2017, pp. 28–40.
[70] J. Serra, *Image Analysis and Mathematical Morphology*. Orlando, FL, USA: Academic, 1983.
[71] R. Klette and A. Rosenfeld, *Digital Geometry*. San Francisco, CA, USA: Morgan Kaufmann, 2003.
[72] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An accurate O(n) solution to the PnP problem," *Int. J. Comput. Vis.*, vol. 81, no. 2, pp. 155–166, 2009.
[73] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 404–417.
[74] H.-J. Chien and R. Klette, "Regularised energy model for robust monocular egomotion estimation," in *Proc. Int. Joint Conf. Comput. Vis. Imag. Comput. Graph., Theory Appl.*, vol. 6, 2011, pp. 361–368.
[75] K. A. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quart. Appl. Math.*, vol. 2, no. 2, pp. 164–168, 1944.
[76] *OpenMP*. Accessed: Nov. 25, 2018. [Online]. Available: www.openmp.org/mp-documents/openmp-4.5.pdf
[77] S. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
[78] H. Kaiming, G. Georgia, D. Piotr, and G. Ross, "Mask R-CNN," *CoRR*, vol. abs/1703.06870, 2017.
[79] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. CVPR*, 2014, pp. 580–587.
[80] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, Apr. 2013.
[81] R. Girshick, "Fast R-CNN," in *Proc. ICCV*, 2015, pp. 1440–1448.
[82] R. Shaoqing, H. Kaiming, G. Ross, and S. Jian, "Faster R-CNN: Towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015.
[83] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2018, pp. 770–778.
[84] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. CVPR*, vol. 1, no. 2, 2017, p. 4.
[85] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. ECCV*, 2014, pp. 740–755.
[86] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," 2016, *arXiv:1612.08242*. [Online]. Available: https://arxiv.org/abs/1612.08242
[87] J. Hui, "mAP (mean average precision) for object detection," Tech. Rep., Mar. 2018. [Online]. Available: https://medium.com/
[88] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nesic, X. Wang, and P. Westling, "High-resolution stereo datasets with subpixel-accurate ground truth," in *Proc. Int. Conf. GCPR*, in Lecture Notes in Computer Science, vol. 8753, 2014, pp. 31–42.
[89] T. Vaudrey, C. Rabe, R. Klette, and J. Milburn, "Differences between stereo and motion behavior on synthetic and real-world stereo sequences," in *Proc. Int. Conf. Image Vis. Comput. New Zealand*, 2008, pp. 1–6.
[90] R. Guzmán, J.-B. Hayet, and R. Klette, "Towards ubiquitous autonomous driving: The CCSAD dataset," in *Proc. Int. Conf. Comput. Anal. Images Patterns*, 2015, pp. 582–593.
[91] A. Börner *et al.*, "IPS—A vision-aided navigation system," in *Proc. Adv. Opt. Technol.*, vol. 6, no. 2, pp. 121–129, 2017.
[92] D. Grießbach, D. Baumbach, and S. Zuev, "Stereo-vision-aided inertial navigation for unknown indoor and outdoor environments," in *Proc. Indoor Positioning Indoor Navigat.*, 2014, pp. 709–716.
[93] S. Nienaber, M. J. Booysen, and R. S. Kroon, "Detecting potholes using simple image processing techniques and real-world footage," in *Proc. Southern Afr. Transp. Conf.*, Jul. 2015.
[94] *PNW Dataset*. Accessed: May 25, 2019. [Online]. Available: www.youtube.com/watch?v=BQo87tGRM74

Amita Dhiman received the master's degree in computer science. She is currently pursuing the Ph.D. degree with the Auckland University of Technology. Teaching Assistant with the Auckland University of Technology. She has coauthored papers in image processing, stereo vision, and deep learning.

Reinhard Klette is a Professor with Auckland University of Technology. He has coauthored more than 300 publications in peer-reviewed journals or conferences and books on computer vision, image processing, geometric algorithms, and panoramic imaging. He is a fellow of the Royal Society of New Zealand. He is on the Honorary Board of the *International Journal of Computer Vision*. He was an Associate Editor of the IEEE PAMI from 2001 to 2008.