sciendo

JAISCR, 2024, Vol. 14, No. 1, pp. 25 – 43
10.2478/jaiscr-2024-0002

BENDING PATH UNDERSTANDING BASED ON ANGLE PROJECTIONS IN FIELD ENVIRONMENTS

Luping Wang[1,*] and Hui Wei[2]

[1]*Laboratory of 3D Scene Understanding and Visual Navigation, School of Mechanical Engineering, University of Shanghai for Science and Technology, Shanghai, 200093, China*

[2]*Laboratory of Algorithms for Cognitive Models, School of Computer Science, Fudan University, Shanghai, 201203, China*

[*]*E-mail: 15110240007@fudan.edu.cn*

Submitted: 2nd January 2023; Accepted: 6th November 2023

Abstract

Scene understanding is a core problem for field robots. However, many unsolved problems, like understanding bending paths, severely hinder the implementation due to varying illumination, irregular features and unstructured boundaries in field environments. Traditional three-dimensional(3D) environmental perception from 3D point clouds or fused sensors are costly and account poorly for field unstructured semantic information. In this paper, we propose a new methodology to understand field bending paths and build their 3D reconstruction from a monocular camera without prior training. Bending angle projections are assigned to clusters. Through compositions of their sub-clusters, bending surfaces are estimated by geometric inferences. Bending path scenes are approximated bending structures in the 3D reconstruction. Understanding sloping gradient is helpful for a navigating mobile robot to automatically adjust their speed. Based on geometric constraints from a monocular camera, the approach requires no prior training, and is robust to varying color and illumination. The percentage of incorrectly classified pixels were compared to the ground truth. Experimental results demonstrated that the method can successfully understand bending path scenes, meeting the requirements of robot navigation in an unstructured environment.

Keywords: bending path, field robots, scene understanding

1 Introduction

Delivery using field robots for emergency supplies is a potential way to avoid dangerous and risky factors. However, there are a multitude of unsettled issues. A major difficulty is understanding unstructured bending paths in field environments, which suffers varying illumination, diverse colours, and unpredictable materials. To understand them through monocular vision can highly improve the terrain adaptability and the working efficiency of vision-based field robots.

Traditional algorithms for understanding scenes from 3D point clouds or RGB-D data are memory intensive and energy-consuming. To understand an unstructured environment through a low-cost monocular camera has advantages in efficiency and

consumption over them. Depth perception without additional information is innate, and explainable visual cues of surfaces with different orientations from monocular vision are of considerable importance to approximate a scene [1, 2, 3]. Compared to precise absolute depth, the relative depth and orientation of different spatial structures are more helpful for scene recovery. Unlike structured indoor and urban scenes, a bending path, which includes many unstructured ropes, planks and fences, appears to be bending structures due to variations and uneven force distribution. What is more, illumination variations in a natural environment make most edges of those structures difficult to shape integrated forms. Their geometric configurations have two-dimensional(2D) angle projections with diverse patterns. Surfaces in a bending path scene often are not so aligned, and these bending angle projections help us to understand such scenes.

In this paper, we propose a new methodology to understand and reconstruct bending path scenes from a low-cost monocular camera without prior training. New clusters of bending angle projections are extracted. Through geometric inferences, compositions of bending surfaces are estimated. Combined with Manhattan structures, bending path scenes are understood through relative pose of different planes in 3D reconstruction. According to the understanding of bending paths, it is possible for a robot to automatically regulate the speed so as to adapt current bending surfaces with sloping gradient.

Unlike deep learning-based methods, the proposed method requires no prior training. Furthermore, the method can understand bending paths without knowledge of the camera's intrinsic parameters nor of precise depth, which enjoys strong market advantages over methods on 3D point clouds or RGB-D data. Simple geometric inferences can account for changes in color and illumination, making applications more reliable and practical in a unstructured environment.

The percentage of incorrectly classified pixels were measured by comparing estimated bending paths to the ground truth. Experimental results indicated that the presented methodology of geometric inferences has bright application prospects in understanding bending path scenes from a resource-constrained vision-based robot.

2 Related work

A number of methods for reconstructing scenes were proposed such as SFM [4] and visual SLAM [5]. A Fused Feature Point Network based on deep learning was presented to learn on raw point clouds exploiting features of global shape [6]. A tracking model was introduced to address the problem of occlusions within large groups of featureless objects through 3D point clouds [7]. However, 3D point clouds are energy-consuming, and have difficulty in explaining geometric cues such as edges, textures and surfaces.

Technology of modelling scenes has progressed greatly. Indoor layouts and scene details can be understood by projections of spatial rectangles [8]. The interaction between indoor objects and layouts with Manhattan assumption were introduced [9]. In addition, another geometric model was introduced for estimating indoor attic surfaces [10]. But they discussed only dominant directions with Manhattan assumption. The camera orientation was taken into consideration by nonlinear Bayesian filtering [11]. However, due to lack of modeling bending surfaces, these methods have difficulty in understanding bending paths.

Deep learning-based algorithms have become popular. A method was proposed to train segmentation models through classification information in the training process of network [12]. Another approach was presented for semi-supervised semantic segmentation to learn from limited pixel-wise annotated samples [13]. A role of knowledge transfer between deep-learning-based classifiers for different crop types was introduced to reduce the retraining time and labeling efforts required for a new crop [14]. Another integrated framework based on CNNs was contended to automate detection/recognition of lesions from in-field images containing part of the coffee tree [15]. However, lack of interpretability make these data-driven algorithms less reliable in understanding a scenes.

Many works were proposed to infer scene semantic information. An algorithm was presented to illuminate the vulnerability aspects of CNNs for semantic segmentation with respect to adversarial attacks [16]. Semantic segmentation was also provided by geometric slanted model and a deep framework through residual connections and fac-

torized convolutions [17, 18]. Another architecture of visual attention was contended to infer objects and curved structures [19, 20]. A multi-task instance segmentation model based on deep neural network was contended to predict road lane and road participants [21]. Another ladder-style DenseNet-based architecture was contended to features high modelling power, efficient upsampling, and inherent spatial efficiency [22]. But these algorithms are prone to failure in understanding bending path scenes due to lack of explaining various bending surfaces.

Impressive researches have been concerned about off-road environmental semantic segmentation and object recognition [23, 24]. For unstructured scenes, a method was proposed for detecting vanishing points in natural scenes, which combines contour-based edges for vanishing points [25]. A semantic segmentation algorithm, based on resolution pyramids and fusion of heterogeneous features, was presented to accomplish dense recognition [26]. Semantic segmentation in an unstructured scene was tried by stereoscopic visual odometry and a contrastive learning model in the DeepLabv3+ framework [27, 28]. To overcome class imbalance and varying environmental topography, a pooled class semantic segmentation was proposed obtain different classes such as sky and obstacles in off-road environments [29]. However, these models account poorly for unstructured boundaries and irregular features of field irrigation ditches.

There are studies on obstacle avoidance and traversable areas for autonomous navigation. Pure geometric models were established to detect the walkable floor and avoid the obstacles on the basis of their shapes and postures [30, 31, 32]. Traversability maps of an unstructured environment were acquired and best path were estimated based on the specific robot characteristics [33, 34]. Other architectures were presented to robustly learn the distribution of traversability costs and the actual road characteristics [35, 36]. Most of these methods just focus on classifying traversable or non-traversable areas, so they are incapable of understanding irrigation ditches with soil and vegetation.

A mapping approach on RGB or RGB-D videos was contended to separately reconstruct 3D models of fruit trees from both sides [37]. Another semantic mapping system for all terrain vehicles (ATVs) was proposed to offer the environmental representation from LiDAR and cameras [38]. A spatial sensing model of multicamera system was built by using panoramic vision, in order to improve the localization precision in weak dynamic environments [39]. However, these approaches don't have the capacity to interpret features of rough ditches in a 3D reconstruction.

Therefore, a methodology was necessary to understand bending paths in unstructured environments that included both Manhattan structures and bending surfaces without prior training from a single low-cost camera. Furthermore, a model of high efficiency and low consumption was needed to meet the requirements of field robots in an unstructured environment.

3 Understanding a bending path

There are many ropes, planks, stones and fences in a bending path scene. Their spatial corners are projected into 2D angle projections with diverse configurations, which can be considered as bending angle projections. Figure 1 shows our system architecture. The input is a monocular capture. Layout vanishing points (LVP) represents the VPs of layout satisfying the Manhattan assumption, while bending structures are interpreted in the following sections.

3.1 Preprocessing

Lines and VPs are extracted [40, 8] as follows:

$$line_i = [x_1^i, y_1^i, x_2^i, y_2^i],\ i \in \mathbb{N}, \tag{1}$$

$$LVP = [MVP; VP_2; VP_3;]; \tag{2}$$

where $\mathbb{N}$ is the number of lines. The main VP (MVP) is a point which is closest to the centre of capture and VP_3 is a vertical one.

An angle projection $\mathbb{A}$ can be considered as a composition of two lines:

$$\mathbb{A} = \{line_s, line_n\}. \tag{3}$$

Here $line_s$ and $line_n$ represent two lines that compose a 2D angle projection [31].

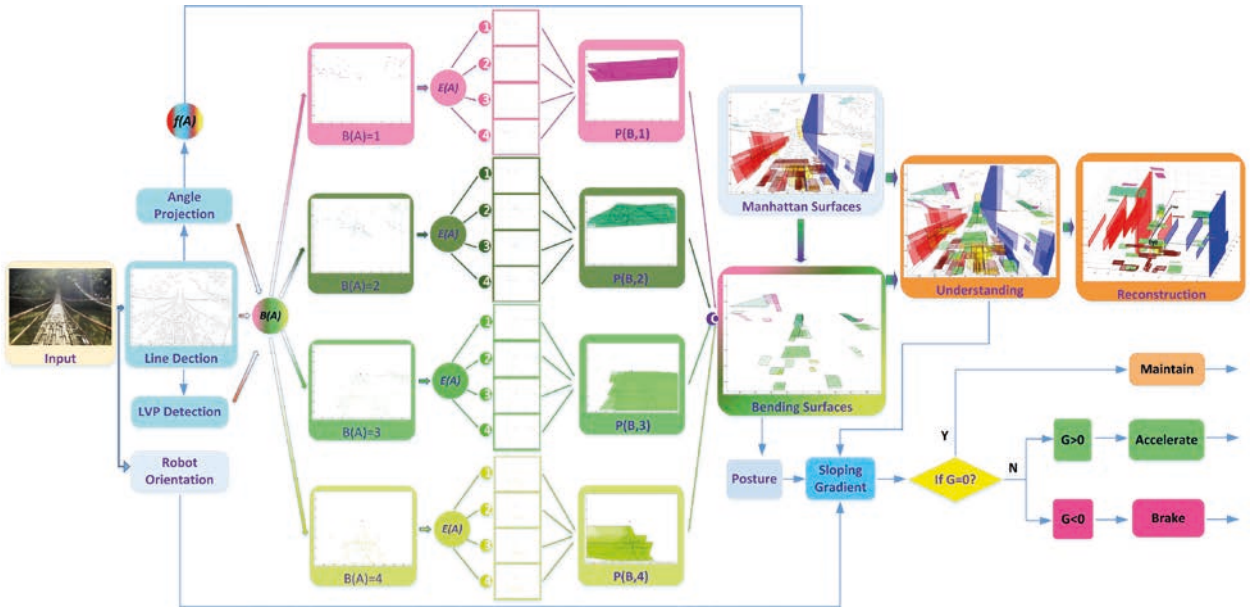

Figure 1. Our system architecture. For a monocular capture, angle projections and *LVP* are extracted based on detected lines. Through new clusters of bending angle projections, sub-clusters, and bending surfaces, it is possible to understand a path scene including bending sloping gradient, which helps robots automatically adjust their speed.

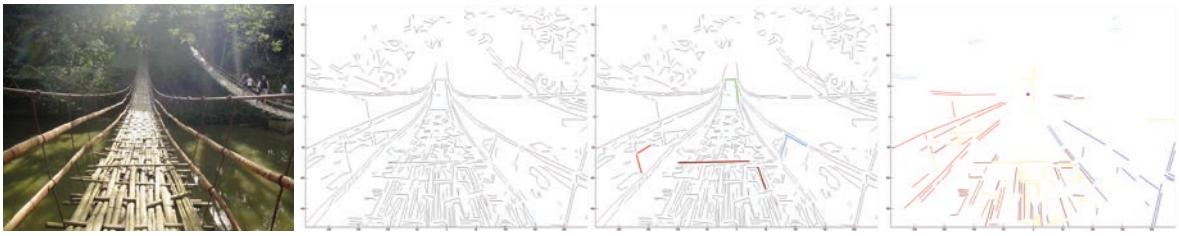

Figure 2. Preprocessing. First: input capture. Second: lines. Third: angle projection examples. Right: Manhattan angle projections.

For an angle projection, the following can be found:

$$\beta_{n_1} = atan((y_n - y_c)/(x_n - x_c)) \quad (4)$$

$$\beta_{n_2} = atan((y_s - y_c)/(x_s - x_c)) \quad (5)$$

$$\beta_{MVP} = atan((y_{MVP} - y_c)/(x_{MVP} - x_c)) \quad (6)$$

$$\beta_{VP_2} = atan((y_{VP_2} - y_c)/(x_{VP_2} - x_c)) \quad (7)$$

$$\beta_{VP_3} = atan((y_{VP_3} - y_c)/(x_{VP_3} - x_c)) \quad (8)$$

$[x_c, y_c]$ is the intersection of an angle projection $\mathbb{A}$. $[x_s, y_s]$ and $[x_n, y_n]$ are midpoints of $line_s$ and $line_n$, respectively. $MVP = (x_{MVP}, y_{MVP})$, $VP_2 = (x_{VP_2}, y_{VP_2})$, and $VP_3 = (x_{VP_3}, y_{VP_3})$. Accordingly, Manhattan angle projections are extracted [31], as shown in Figure 2.

3.2 Bending angle projections

For a bending angle projection, its two lines are not converging to LVP. Hence, a model to extract new clusters of bending angle projections is founded. As to angles β_{MVP} and β_{VP_3}, the minimum and maximum can be found,

$$\widetilde{\phi} = min(\beta_{MVP}, \beta_{VP_3}) \quad (9)$$

$$\widetilde{\Phi} = max(\beta_{MVP}, \beta_{VP_3}) \quad (10)$$

A bending angle projection can be seen as ones that are transformed by rotating a Manhattan angle projection around its line that is converging to VP_2, which should satisfy the following constraints:

$$\beta_{n_2} \to \beta_{VP_2} \quad (11)$$

Assuming that $B(\mathbb{A})$ is a new cluster of bending angle projections, their geometric classification rules are interpreted as shown in Table 1. According to geometric constraints, new clusters of bending angle projections are marked as $B(\mathbb{A}) = 1,2,3,4$, which represent the different position and orientation, respectively.

According to geometric constraints in Table 1, it is possible to extract new clusters of bending angle projections ($\{\mathbb{A}^{B_1}\}$, $\{\mathbb{A}^{B_2}\}$, $\{\mathbb{A}^{B_3}\}$, and $\{\mathbb{A}^{B_4}\}$), as shown in Figure 3.

Table 1. Classifying bending angle projections.

c_1	c_2	c_3	$B(\mathbb{A})$
$x_c > x_{MVP}$	$y_c > y_{MVP}$	$\beta_{n_1} \notin (\widetilde{\phi}, \widetilde{\Phi})$	1
$x_c < x_{MVP}$	$y_c > y_{MVP}$	$\beta_{n_1} \in (\widetilde{\phi}, \widetilde{\Phi})$	1
$x_c > x_{MVP}$	$y_c > y_{MVP}$	$\beta_{n_1} \in (\widetilde{\phi}, \widetilde{\Phi})$	2
$x_c < x_{MVP}$	$y_c > y_{MVP}$	$\beta_{n_1} \notin (\widetilde{\phi}, \widetilde{\Phi})$	2
$x_c < x_{MVP}$	$y_c < y_{MVP}$	$\beta_{n_1} \notin (\widetilde{\phi}, \widetilde{\Phi})$	3
$x_c > x_{MVP}$	$y_c < y_{MVP}$	$\beta_{n_1} \in (\widetilde{\phi}, \widetilde{\Phi})$	3
$x_c > x_{MVP}$	$y_c < y_{MVP}$	$\beta_{n_1} \notin (\widetilde{\phi}, \widetilde{\Phi})$	4
$x_c < x_{MVP}$	$y_c < y_{MVP}$	$\beta_{n_1} \in (\widetilde{\phi}, \widetilde{\Phi})$	4

3.3 Sub-clusters

A Manhattan rectangular surface can be reshaped by the Manhattan angle projection and LVP. However, since two lines of an angle projection are not converging to LVP, it is impossible to rebuilt a bending rectangular plane by LVP and the corresponding bending angle projection. Based on above geometric constraints, a bending surface can be considered as compositions of bending angle projections. Therefore, it is important to infer subclusters of bending angle projections.

A bending angle projection can be defined:

$$\mathbb{A}^{B_k} = [line_s^{B_k}, line_n^{B_k}],\ k \in 1,2,3,4 \quad (12)$$

$$p_c^{\mathbb{A}^{B_k}} = \Gamma(line_s^{B_k}, line_n^{B_k}) \quad (13)$$

Here $line_s^{B_k}$ is the line related to β_{n_2}, and $line_n^{B_k}$ is the one corresponding to β_{n_1}. Γ is a function that calculates a intersection of two lines. $p_c^{\mathbb{A}^{B_k}}$ is the intersection of $line_s^{B_k}$ and $line_n^{B_k}$.

For $line_s^{B_k}$, the following vectors can be found:

$$\gamma_s^{\mathbb{A}^{B_k}} = arccos \frac{(p_m^{line_s^{B_k}} - p_c^{\mathbb{A}^{B_k}})(VP_2 - p_c^{\mathbb{A}^{B_k}})}{|p_m^{line_s^{B_k}} - p_c^{\mathbb{A}^{B_k}}||VP_2 - p_c^{\mathbb{A}^{B_k}}|} \quad (14)$$

$p_m^{line_s^{B_k}}$ means the midpoint of $line_s^{B_k}$, and $\gamma_s^{\mathbb{A}^{B_k}}$ is a geometric angle with range $[0, \pi]$.

For $line_n^{B_k}$, the following is founded:

$$\gamma_n^{\mathbb{A}^{B_k}} = [x^{\gamma_n^{\mathbb{A}^{B_k}}}, y^{\gamma_n^{\mathbb{A}^{B_k}}}] = p_m^{line_n^{B_k}} - p_c^{\mathbb{A}^{B_k}} \quad (15)$$

Here $p_m^{line_n^{B_k}}$ means the midpoint of $line_n^{B_k}$.

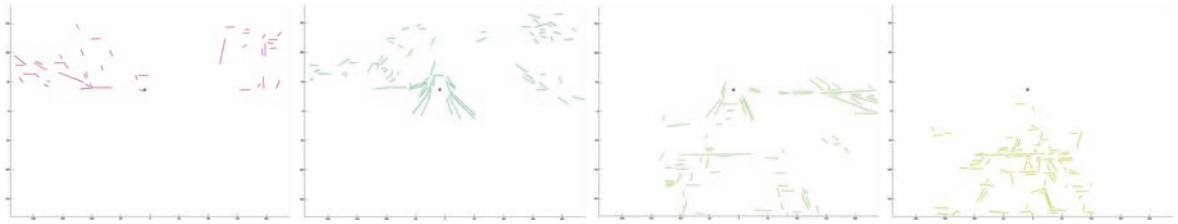

Figure 3. Bending angle projections. First to fourth: four sets of bending angle projections $\{\mathbb{A}^{B_1}\}$, $\{\mathbb{A}^{B_2}\}$, $\{\mathbb{A}^{B_3}\}$, and $\{\mathbb{A}^{B_4}\}$, respectively. Based on explainable geometric constraints in Table 1, bending angle projections are successfully extracted.

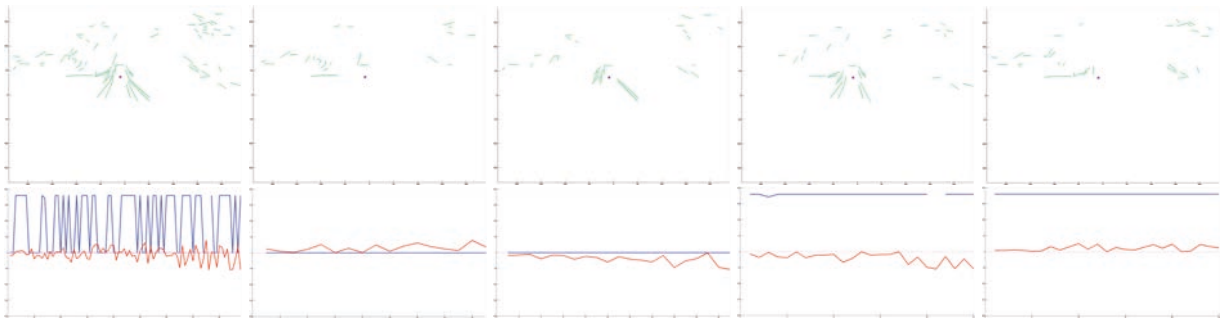

Figure 4. An example of extracting sub-clusters of bending angle projections $\{\mathbb{A}^{B_2}\}$. Top row: first: bending angle projections $\{\mathbb{A}^{B_2}\}$; second to fifth: sub-clusters $E(\mathbb{A}^{B_2}) = 1,2,3,4$, respectively. Bottom row: first: $\gamma_s^{\mathbb{A}^{B_2}}$ is blue line and $\gamma_n^{\mathbb{A}^{B_2}}$ is red line for bending angle projections when $B(\mathbb{A}) = 2$; second to fifth: $\gamma_s^{\mathbb{A}^{B_2}}$ and $\gamma_n^{\mathbb{A}^{B_2}}$ for each sub-cluster when $E(\mathbb{A}^{B_2}) = 1,2,3,4$, respectively. Through geometric constraints in Table 2, sub-clusters are extracted.

Table 2. Sub-cluster of bending angle projections.

c_1	c_2	$E(\mathbb{A}^{B_k})$
$\gamma_s^{\mathbb{A}^{B_k}} \to 0$	$y\gamma_n^{\mathbb{A}^{B_k}} > 0$	1
$\gamma_s^{\mathbb{A}^{B_k}} \to 0$	$y\gamma_n^{\mathbb{A}^{B_k}} < 0$	2
$\gamma_s^{\mathbb{A}^{B_k}} \to \pi$	$y\gamma_n^{\mathbb{A}^{B_k}} < 0$	3
$\gamma_s^{\mathbb{A}^{B_k}} \to \pi$	$y\gamma_n^{\mathbb{A}^{B_k}} > 0$	4

Each cluster of bending angle projections ($B(\mathbb{A}) = 1,2,3,4$) are assigned to four sub-clusters by geometric constraints as shown in Table 2. Through analyzing $\gamma_s^{\mathbb{A}^{B_k}}$ and $\gamma_n^{\mathbb{A}^{B_k}}$ for bending angle projections, it is possible to extract sub-clusters $E(\mathbb{A}^{B_k}) = 1,2,3,4$. For example, four sub-clusters of bending angle projections $\{\mathbb{A}^{B_2}\}$ are shown in Figure 4.

3.4 Sub-cluster-pair

Due to complicated variation of occlusion and illumination in a natural scene, textures always appear to be patchy and faint, resulting in fragmentary surfaces with scattered angle projections. Therefore, it is important to reshape these surfaces through composing scattered angle projections. A surface can be seen as a composition of two bending angle projections from different sub-clusters, which can be considered as a sub-cluster-pair.

Assuming that a bending angle projection $\mathbb{A}^{B_k}$ has been clustered by $E(\mathbb{A}^{B_k})$, for example, $\mathbb{A}^{B_1,E_3}$ means an angle projection in $B(\mathbb{A}) = 1$ and $E(\mathbb{A}^{B_1}) = 3$.

Then the following can be found,

$$\Pi^{B_k,E_1,E_3} = \begin{bmatrix} \mathbb{A}^{B_k,E_1} \\ \mathbb{A}^{B_k,E_3} \end{bmatrix} = \begin{bmatrix} line_s^{B_k,E_1}, line_n^{B_k,E_1} \\ line_s^{B_k,E_3}, line_n^{B_k,E_3} \end{bmatrix} \quad (16)$$

$$= \begin{bmatrix} \Gamma(line_s^{B_k,E_1}, line_n^{B_k,E_1}) \\ \Gamma(line_s^{B_k,E_3}, line_n^{B_k,E_1}) \\ \Gamma(line_s^{B_k,E_3}, line_n^{B_k,E_3}) \\ \Gamma(line_s^{B_k,E_1}, line_n^{B_k,E_3}) \end{bmatrix} \quad (17)$$

$$St.\ E(line_s^{B_k,E_1}, line_n^{B_k,E_3}) = 4 \quad (18)$$

$$E(line_s^{B_k,E_3}, line_n^{B_k,E_1}) = 2 \quad (19)$$

$$\Pi^{B_k,E_2,E_4} = \begin{bmatrix} \mathbb{A}^{B_k,E_2} \\ \mathbb{A}^{B_k,E_4} \end{bmatrix} = \begin{bmatrix} line_s^{B_k,E_2}, line_n^{B_k,E_2} \\ line_s^{B_k,E_4}, line_n^{B_k,E_4} \end{bmatrix} \quad (20)$$

$$= \begin{bmatrix} \Gamma(line_s^{B_k,E_4}, line_n^{B_k,E_2}) \\ \Gamma(line_s^{B_k,E_2}, line_n^{B_k,E_2}) \\ \Gamma(line_s^{B_k,E_2}, line_n^{B_k,E_4}) \\ \Gamma(line_s^{B_k,E_4}, line_n^{B_k,E_4}) \end{bmatrix} \quad (21)$$

$$St.\ E(line_s^{B_k,E_2}, line_n^{B_k,E_4}) = 3 \quad (22)$$

$$E(line_s^{B_k,E_4}, line_n^{B_k,E_2}) = 1 \quad (23)$$

Here $k \in 1,2,3,4$. Π^{B_k,E_1,E_3} is a surface composed by two angle projections $\mathbb{A}^{B_k,E_1}$ and $\mathbb{A}^{B_k,E_3}$. It also can be described as four lines (Eq. 16) and four vertexes (Eq.17). Π^{B_k,E_2,E_4} means a surface of $\mathbb{A}^{B_k,E_2}$ and $\mathbb{A}^{B_k,E_4}$. Π^{B_k,E_1,E_3} and Π^{B_k,E_2,E_4} are two types of sub-cluster-pair. E_1,E_2,E_3,E_4 are four sub-clusters. Assuming that $\{\Pi^{B_k,E_1,E_3}$ and $\{\Pi^{B_k,E_2,E_4}\}$ are 2 sets of these two sub-cluster-pair types, the following can be founded,

$$\mathbb{P}(B,k) = \{\{\Pi^{B_k,E_1,E_3}\};\{\Pi^{B_k,E_2,E_4}\}\},\ k \in 1,2,3,4 \quad (24)$$

Here $\mathbb{P}(B,k)$ (e.g., $\mathbb{P}(B,1)$, $\mathbb{P}(B,2)$, $\mathbb{P}(B,3)$, and $\mathbb{P}(B,4)$) are four surfaces sets estimated by sub-cluster-pairs, as shown in Figure 5.

3.5 Bending surfaces

Bending structures can be seen as a combination of potential surfaces stemming from different clusters. Assuming that $\Pi(B,i)$ and $\Pi(B,j)$ are two surfaces from two sets $\mathbb{P}(B_i)$ and $\mathbb{P}(B_j)$, respectively, four vertexes of one surface can be calculated, and these vertexes obviously belong to different sub-clusters.

For example, four points $[p_{E_1}^{B_i}, p_{E_2}^{B_i}, p_{E_3}^{B_i}, p_{E_4}^{B_i}]$ represent vertexes of $\Pi(B_i)$, and $\iota_{E_1,E_2}^{B_i} = [p_{E_1}^{B_i}, p_{E_2}^{B_i}]$ means it is a line composed by two vertexes. Then a combination and its geometric constraints can be

Figure 5. Bending surfaces of combination of sub-cluster-pairs. $\mathbb{P}(B,k)$ represents a set of sub-cluster-pairs from bending clusters. For two $\mathbb{P}(B,k)$, it is possible to obtain corresponding combinations satisfying geometric constraints. Those sub-cluster-pairs that do not satisfy the geometric constraints will be eliminated, with no combinations for such two $\mathbb{P}(B,k)$, e.g., $\mathbb{P}(B,1)$ and $\mathbb{P}(B,4)$.

found as follows,

$$\mathbb{C}^{i,j} = \begin{bmatrix} \Pi(B_i) \\ \Pi(B_j) \end{bmatrix}, \ i,j \in 1,2,3,4 \tag{25}$$

$$= \begin{bmatrix} p_{E_1}^{B_i}, p_{E_2}^{B_i}, p_{E_3}^{B_i}, p_{E_4}^{B_i} \\ p_{E_1}^{B_j}, p_{E_2}^{B_j}, p_{E_3}^{B_j}, p_{E_4}^{B_j} \end{bmatrix} \tag{26}$$

$$= \begin{bmatrix} \iota_{E_1,E_2}^{B_i}, \iota_{E_3,E_4}^{B_i} \\ \iota_{E_1,E_2}^{B_j}, \iota_{E_3,E_4}^{B_j} \end{bmatrix} \tag{27}$$

$$St.\ \tilde{D}(p_{E_1}^{B_i}, p_{E_2}^{B_j}) \to 0 \tag{28}$$

$$\tilde{D}(p_{E_3}^{B_i}, p_{E_4}^{B_j}) \to 0 \tag{29}$$

$$\iota_{B_j}^{B_i} = [\Gamma(\iota_{E_1,E_2}^{B_i}, \iota_{E_3,E_4}^{B_i}), \Gamma(\iota_{E_1,E_2}^{B_j}, \iota_{E_3,E_4}^{B_j})] \to VP_3 \tag{30}$$

Here $\mathbb{C}^{i,j}$ represents a combination of two potential surfaces. $\tilde{D}$ means a function that calculates Euclidean distance of two points. Eq. 28 and Eq. 29 mean that two adjacent rectangular surfaces that satisfy integrity constraints are more likely to compose a combination.

In Eq. 30, $\iota_{B_j}^{B_i}$ means a line composed by two points $\Gamma(\iota_{E_1,E_2}^{B_i}, \iota_{E_3,E_4}^{B_i})$ and $\Gamma(\iota_{E_1,E_2}^{B_j}, \iota_{E_3,E_4}^{B_j})$, which can be considered as orientation constraints. Here Γ is a function that calculates a intersection of two lines. Although angle projections of bending structures have diverse 2D configurations, their lines still keep specific geometric constraints. Here a bending angle projection can be considered as the one that is transformed by rotating an original Manhattan angle projection around its line that is converging to VP_2. Therefore, for two lines of a bending angle projection, one line converges to VP_2, and another line should converge to a point that belongs to a line converging to VP_3, which can be defined as $\iota_{B_j}^{B_i} \to VP_3$.

For all $\mathbb{P}(B,k), k \in 1,2,3,4$, corresponding combinations can be obtained, as shown in Figure 5. For two $\mathbb{P}(B,k)$, those sub-cluster-pairs that do not satisfy the geometric constraints will be eliminated. For example, for two $\mathbb{P}(B,1)$ and $\mathbb{P}(B,4)$, there are no combinations satisfying the geometric constraints. In a bending path, those bending surfaces can be written as $\{\mathbb{C}^{i,j}\}$.

Finally, it is possible to understand a scene including Manhattan planes and bending planes, as shown in Figure 6.

3.6 Sloping gradient

Based on understanding a bending path, a robot should model the sloping gradient of bending surfaces and regulate its speed so as to accomplish navigation. The depth of bending surfaces is difficult to accurately calculate from a single camera, and it is essential to infer the relative relationship between bending surfaces and robot orientation. Because the world coordinate system and the camera coordinate system are not coincident, the approximated bending planes can be regarded as those that are rotated and translated relative to the coordinates.

Assuming that Π^{B_k} is a surface that belongs to a bending combination ($\mathbb{C}^{i,j}$) extracted by the above inference, it concludes two angle projections ($\mathbb{A}_1^{B_k}$ and $\mathbb{A}_2^{B_k}$, $k \in 1,2,3,4$) that can also be described as follows,

$$\Pi^{B_k} = \begin{bmatrix} \mathbb{A}_1^{B_k} \\ \mathbb{A}_2^{B_k} \end{bmatrix} = \begin{bmatrix} line_s^{B_k,1}, line_n^{B_k,1} \\ line_s^{B_k,2}, line_n^{B_k,2} \end{bmatrix} \tag{31}$$

The VP of this surface can be obtained by calculating the follows,

$$\Psi^{\Pi^{B_k}} = \Gamma(line_n^{B_k,1}, line_n^{B_k,2}) \tag{32}$$

Here $\Psi^{\Pi^{B_k}}$ means the VP of Π^{B_k}. $line_n^{B_k,1}$ and $line_n^{B_k,2}$ are two lines that are not converge to LVP in bending angle projections.

Here $\mathbb{A}_1^{B_k}$ and $\mathbb{A}_2^{B_k}$ are belonging to a surface. Assuming that the maximum depth here is $\mathbb{D}$, and $\Psi^{\Pi^{B_k}}$ represents a VP in the bending path, the angle of sloping gradient can be approximately estimated by solving the following:

$$G^{\Psi^{\Pi^{B_k}}} = atan(\frac{y_{\Psi^{\Pi^{B_k}}}}{\sqrt{\mathbb{D}^2 + x^2_{\Psi^{\Pi^{B_k}}}}}) - atan(\frac{y_M}{\sqrt{\mathbb{D}^2 + x^2_{\Psi^{\Pi^{B_k}}}}}) \tag{33}$$

Here, $G^{\Psi^{\Pi^{B_k}}}$ is the sloping angle of rotation in relation to $\Psi^{\Pi^{B_k}}$. According to the value of sloping gradient, the posture of a bending structure can be estimated.

When a robot faces a bending surface, it is important to determine how to adjust its speed to climb the bending sloped surfaces. For $frame^t$ (a capture at time t), here $G^{\Psi^{\Pi^{B_k}}}(t)$ represents the bending sloped gradient between the robot orientation and the bending surface at time t. In this way, the acceleration rules of a robot in the camera coordinate system are as shown in Table 3 (e.g., here the estimated rotation is related to $\Psi^{\Pi^{B_k}}$).

Table 3. Acceleration rules for estimating rotation related to $\Psi^{\Pi^{B_k}}$.

Sloping gradient	Posture	Motion
$G^{\Psi^{\Pi^{B_k}}}(t) < 0$	$down-sloping$	$Brake$
$G^{\Psi^{\Pi^{B_k}}}(t) > 0$	$up-sloping$	$Accelerate$
$G^{\Psi^{\Pi^{B_k}}}(t) \to 0$	$horizontal$	$Maintain$

Based on an understanding of sloping gradient of bending sloped scenes, the robot can regulate speed according to the posture of a bending surface. The process of speed control can be modeled as a motion of function of $G^{\Psi^{\Pi^{B_k}}}(t)$. With the sloped gradient value, the robot can adjust the acceleration and brake, step by step. Besides, the higher absolute value ($abs(G^{\Psi^{\Pi^{B_k}}}(t))$) indicates it needs more aggressive motion.

3.7 Reconstruction of bending surfaces

For bending angle projections $\mathbb{A}_1^{B_k}$ and $\mathbb{A}_2^{B_k}$, the following can be found,

$$\varsigma_s^{B_k,1} = \frac{2p_m^{line_s^{B_k,1}} + l^{line_s^{B_k,1}}(p_m^{line_s^{B_k,1}} - p_c^{B_k,1})}{2\|p_m^{line_s^{B_k,1}} - p_c^{B_k,1}\|_2} \tag{34}$$

$$\varsigma_n^{B_k,1} = \frac{2p_m^{line_n^{B_k,1}} + l^{line_n^{B_k,1}}(p_m^{line_n^{B_k,1}} - p_c^{B_k,1})}{2\|p_m^{line_n^{B_k,1}} - p_c^{B_k,1}\|_2} \tag{35}$$

$$\varsigma_s^{B_k,2} = \frac{2p_m^{line_s^{B_k,2}} + l^{line_s^{B_k,2}}(p_m^{line_s^{B_k,2}} - p_c^{B_k,2})}{2\|p_m^{line_s^{B_k,2}} - p_c^{B_k,2}\|_2} \tag{36}$$

$$\varsigma_n^{B_k,2} = \frac{2p_m^{line_n^{B_k,2}} + l^{line_n^{B_k,2}}(p_m^{line_n^{B_k,2}} - p_c^{B_k,2})}{2\|p_m^{line_n^{B_k,2}} - p_c^{B_k,2}\|_2} \tag{37}$$

Here $p_c^{B_k,1}$ represents the intersection of $line_s^{B_k,1}$ and $line_n^{B_k,1}$. $p_m^{line_s^{B_k,1}}$ and $l^{line_s^{B_k,1}}$ are the midpoint and length. $\varsigma_s^{B_k,1}$ and $\varsigma_n^{B_k,1}$ mean two points that are further from the intersection. Similarly, $p_c^{B_k,2}$ is the intersection of $line_s^{B_k,2}$ and $line_n^{B_k,2}$. $p_m^{line_s^{B_k,2}}$ and $l^{line_s^{B_k,2}}$ are the midpoint and length, respectively. $\varsigma_s^{B_k,2}$ and $\varsigma_n^{B_k,2}$ are two points that are further from $p_c^{B_k,2}$.

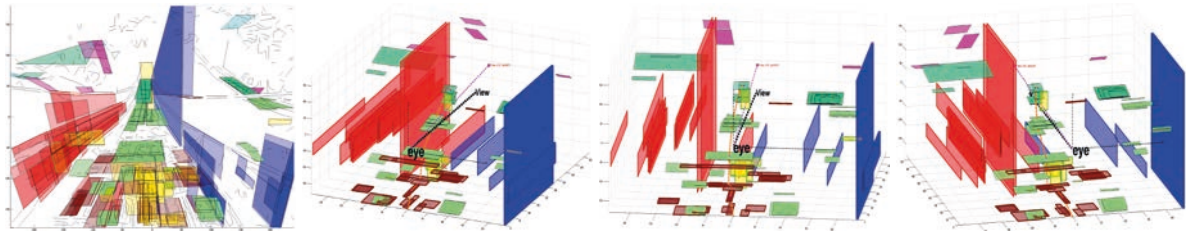

Figure 6. Scene reconstruction. Left: scene understanding. Second to Fourth: scene reconstruction from different perspectives. The viewer is at the position symbolized "eye" looking toward "view". The scene is reconstructed through Manhattan spatial rectangles (e.g., blue, red, brown areas) and bending surfaces (e.g., turquoise and mint areas).

According to Eq. 11, the following can be found:

$$\varsigma^{B_k,1} = \Gamma([\varsigma_n^{B_k,1}, VP_2], [\varsigma_s^{B_k,1}, \Psi^{\Pi^{B_k}}]) \tag{38}$$

$$\varsigma^{B_k,2} = \Gamma([\varsigma_n^{B_k,2}, VP_2], [\varsigma_s^{B_k,2}, \Psi^{\Pi^{B_k}}]) \tag{39}$$

Here $\varsigma^{B_k,1}$ is a intersection of line $[\varsigma_n^{B_k,1}, VP_2]$ and line $[\varsigma_s^{B_k,1}, \Psi^{\Pi^{B_k}}]$. $\varsigma^{B_k,2}$ represents another intersection of line $[\varsigma_n^{B_k,2}, VP_2]$ and line $[\varsigma_s^{B_k,2}, \Psi^{\Pi^{B_k}}]$.

Therefore, vertexes of surfaces of bending angle projections $\mathbb{A}_1^{B_k}$ and $\mathbb{A}_2^{B_k}$ can be expressed,

$$\left[X^{B_k,1}, Y^{B_k,1}\right] = \left[p_c^{B_k,1}; \varsigma_s^{B_k,1}; \varsigma^{B_k,1}; \varsigma_n^{B_k,1};\right]^T \tag{40}$$

$$\left[X^{B_k,2}, Y^{B_k,2}\right] = \left[p_c^{B_k,2}; \varsigma_s^{B_k,2}; \varsigma^{B_k,2}; \varsigma_n^{B_k,2};\right]^T \tag{41}$$

Because accurate size and depth cannot be calculated from a monocular camera, the relative relationship between projected surfaces and VPs must be estimated. Assuming that $\mathbb{D}$ means the maximum depth, and H, W are height and width of a capture, respectively.

$$\hat{z}_{\varepsilon^{B_k,1}} = \begin{cases} \mathbb{D}(1+|\frac{y_M - max(Y^{B_k,1})}{y_M - H/2}|), & k = 1,2 \\ \mathbb{D}(1-|\frac{y_M - min(Y^{B_k,1})}{y_M + H/2}|), & k = 3,4 \end{cases} \tag{42}$$

$$\hat{z}_{\xi^{B_k,1}} = \begin{cases} \mathbb{D}(1+|\frac{y_M - min(Y^{B_k,1})}{y_M - H/2}|), & k = 1,2 \\ \mathbb{D}(1-|\frac{y_M - max(Y^{B_k,1})}{y_M + H/2}|), & k = 3,4 \end{cases} \tag{43}$$

$$\hat{z}_{\varepsilon^{B_k,2}} = \begin{cases} \mathbb{D}(1+|\frac{y_M - max(Y^{B_k,2})}{y_M - H/2}|), & k = 1,2 \\ \mathbb{D}(1-|\frac{y_M - min(Y^{B_k,2})}{y_M + H/2}|), & k = 3,4 \end{cases} \tag{44}$$

$$\hat{z}_{\xi^{B_k,2}} = \begin{cases} \mathbb{D}(1+|\frac{y_M - min(Y^{B_k,2})}{y_M - H/2}|), & k = 1,2 \\ \mathbb{D}(1-|\frac{y_M - max(Y^{B_k,2})}{y_M + H/2}|), & k = 3,4 \end{cases} \tag{45}$$

Here $MVP = [x_M, y_M]$. In this way, 3D reconstruction of bending surfaces $\hat{R}^{B_k,1}$ and $\hat{R}^{B_k,2}$ that is coming from bending angle projections $\mathbb{A}_1^{B_k}$ and $\mathbb{A}_2^{B_k}$ can be estimated:

$$\hat{R}^{B_k,1} = \begin{bmatrix} \hat{X}^{B_k,1} \\ \hat{Y}^{B_k,1} \\ \hat{Z}^{B_k,1} \end{bmatrix}^T = \begin{bmatrix} min(X^{B_k,1}), min(Y^{B_k,1}), \hat{z}_{\varepsilon^{B_k,1}} \\ min(X^{B_k,1}), min(Y^{B_k,1}), \hat{z}_{\xi^{B_k,1}} \\ max(X^{B_k,1}), min(Y^{B_k,1}), \hat{z}_{\xi^{B_k,1}} \\ max(X^{B_k,1}), min(Y^{B_k,1}), \hat{z}_{\varepsilon^{B_k,1}} \end{bmatrix} \tag{46}$$

$$\hat{R}^{B_k,2} = \begin{bmatrix} \hat{X}^{B_k,2} \\ \hat{Y}^{B_k,2} \\ \hat{Z}^{B_k,2} \end{bmatrix}^T = \begin{bmatrix} min(X^{B_k,2}), min(Y^{B_k,2}), \hat{z}_{\varepsilon^{B_k,2}} \\ min(X^{B_k,2}), min(Y^{B_k,2}), \hat{z}_{\xi^{B_k,2}} \\ max(X^{B_k,2}), min(Y^{B_k,2}), \hat{z}_{\xi^{B_k,2}} \\ max(X^{B_k,2}), min(Y^{B_k,2}), \hat{z}_{\varepsilon^{B_k,2}} \end{bmatrix} \tag{47}$$

As two bending angle projections $\mathbb{A}_1^{B_k}$ and $\mathbb{A}_2^{B_k}$ belong to a surface, the angle of rotation can be approximately estimated by solving the following:

$$\rho = atan(x_M/\mathbb{D}) \tag{48}$$

$$\rho^{B_k,1} = \rho^{B_k,2} = G^{\Psi^{\Pi^{B_k}}} \tag{49}$$

Here, ρ is the horizontal angle of rotation related to LVP. $\rho^{B_k,1}$ and $\rho^{B_k,2}$ are angles of rotation in relation to $\Psi^{\Pi^{B_k}}$. So bending surfaces for these two angle projections in 3D reconstruction are approxi-

mated:

$$\Lambda^{B_k,1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\rho^{B_k,1} & -sin\rho^{B_k,1} \\ 0 & sin\rho^{B_k,1} & cos\rho^{B_k,1} \end{bmatrix} \begin{bmatrix} cos\rho & 0 & -sin\rho \\ 0 & 1 & 0 \\ sin\rho & 0 & cos\rho \end{bmatrix} \quad (50)$$

$$\Lambda^{B_k,2} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\rho^{B_k,2} & -sin\rho^{B_k,2} \\ 0 & sin\rho^{B_k,2} & cos\rho^{B_k,2} \end{bmatrix} \begin{bmatrix} cos\rho & 0 & -sin\rho \\ 0 & 1 & 0 \\ sin\rho & 0 & cos\rho \end{bmatrix} \quad (51)$$

$$\hat{R}^{B_k,1,*} = \begin{bmatrix} \hat{X}^{B_k,1} - X_0^{B_k,1} \\ \hat{Y}^{B_k,1} - Y_0^{B_k,1} \\ \hat{Z}^{B_k,1} - Z_0^{B_k,1} \end{bmatrix}^T \Lambda^{B_k,1} + \begin{bmatrix} X_0^{B_k,1} \\ Y_0^{B_k,1} \\ Z_0^{B_k,1} \end{bmatrix}^T \quad (52)$$

$$\hat{R}^{B_k,2,*} = \begin{bmatrix} \hat{X}^{B_k,2} - X_0^{B_k,2} \\ \hat{Y}^{B_k,2} - Y_0^{B_k,2} \\ \hat{Z}^{B_k,2} - Z_0^{B_k,2} \end{bmatrix}^T \Lambda^{B_k,2} + \begin{bmatrix} X_0^{B_k,2} \\ Y_0^{B_k,2} \\ Z_0^{B_k,2} \end{bmatrix}^T \quad (53)$$

Here, $[X_0^{B_k,1}, Y_0^{B_k,1}, Z_0^{B_k,1}]$ and $[X_0^{B_k,2}, Y_0^{B_k,2}, Z_0^{B_k,2}]$ are centre points of $\hat{R}^{B_k,1}$ and $\hat{R}^{B_k,2}$, respectively. $\Lambda^{B_k,1}$ and $\Lambda^{B_k,2}$ are corresponding orientation matrixes for $\hat{R}^{B_k,1}$ and $\hat{R}^{B_k,2}$. $\hat{R}^{B_k,1,*}$ and $\hat{R}^{B_k,2,*}$ are 3D reconstruction for two surfaces of bending angle projections. As shown in Figure 6, combined with Manhattan surfaces, a scene can be understood in 3D reconstruction.

4 Experimental results

4.1 Experimental setup and datasets

Our study is entirely geometric inference-based, and enables understanding bending pathways in various field environments using a low-cost monocular camera alone. Since no precise depth data or any prior training are needed, our experiments were performed on a computer with an Intel Core i7-6500 2.50 GHz CPU with 2 GB of RAM and without an additional high-performance GPU.

There are a variety of datasets for understanding scenes, including widespread KITTI [41] and Cityscapes [42] datasets, both of which are structured city driving datasets; the outside unstructured environment (e.g., TAS500 [43], FDWW [8], RUGD [44] and SUN [45]). There are diverse bending pathways in field environments in the FDWW dataset [8]. Our approach is designed to address understanding of bending paths in a field environment; hence, the FDWW [8] and SUN [45] datasets involving diverse field bending paths were adopted to assess the validity of the proposed approach.

4.2 Evaluation

We present an interpretable model to understand a scene including bending paths without prior training or any precise depth data. There are diverse bending paths and roads in unstructured environments in FDWW dataset [8]. Regarding a bending rope bridge scene from FDWW dataset [8], as shown in Figure 7, our method is able to understand sloping and bending scenarios. The estimated bending path was compared to the ground truth, the percentage of pixels incorrectly classified was evaluated in terms of different scenes, as shown in Table 4. It calculates the incorrectly classified pixels, i.e., $error = \frac{FP+FN}{TP+FP+FN}$, where TP, FP, and FN are the numbers of true positive, false positive, and false negative pixels, respectively. Compared with other traditional methods, as shown in Figure 10, the experimental results demonstrate that the proposed method has advantages in understanding bending scenarios, which is more appropriate for a visual-based navigation.

Table 4. Error of understanding field bending pathways on FDWW dataset [8].

Method	Pathway	Side
H.W. [8]	52.3%	32.8%
Wang [32]	28.6%	29.6%
Our method	9.3%	22.3%

Field bending rope ways from the SUN database [45] were also evaluated. The proposed approach can not only infer the side ropes but can understand bending walkways. Comparing the estimated pathway to the ground truth, the evaluated percentage of incorrectly classified pixels is shown in Table 5.

Table 5. Average error of understanding bending rope ways on the SUN dataset [45].

Method	Error
H.W. [8]	38.6%
Our method	12.8%

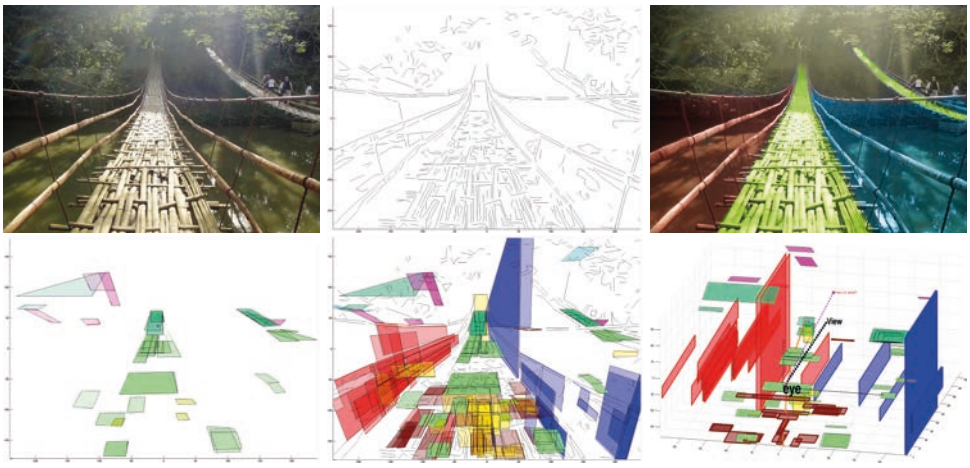

Figure 7. Scene understanding. Top left: input image [8]. Top middle: lines. Top right: ground truth. Bottom left: bending surfaces. Bottom middle: scene understanding. Bottom right: scene reconstruction. Compared to the ground truth, structures (e.g., blue, red, brown areas) and bending surfaces (e.g., turquoise and mint areas) can be understood without any prior training.

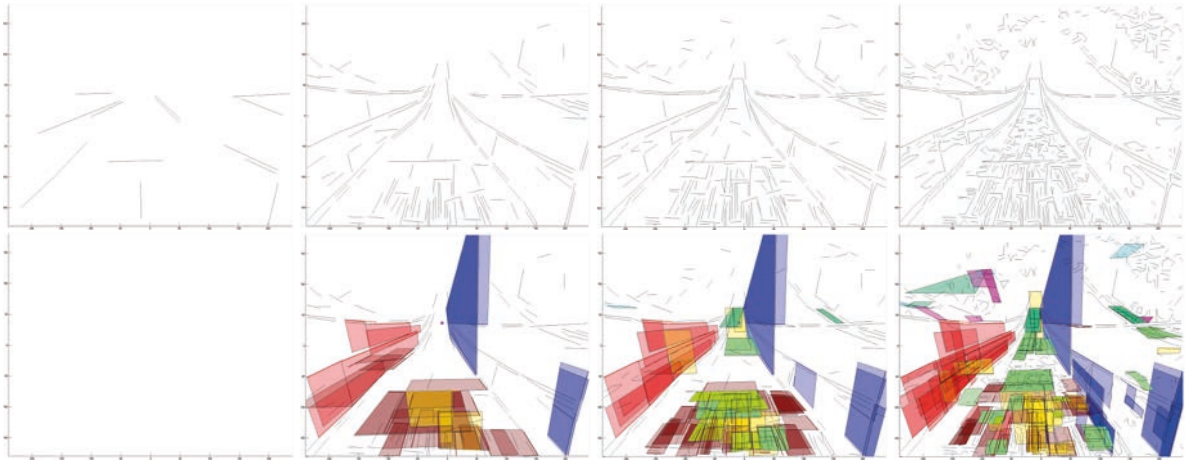

Figure 8. A study for the sensitivity of extracting lines. Tow row: extracted lines with increasing λ_l. Bottom row: corresponding scene understanding. With no extracted lines (first column), a scene is hard to understand. Through more extracted lines, the proposed algorithm can afford more scene details.

4.3 Ablation studies

In this paper, the presented algorithm develops a geometric inference to understand bending paths in unstructured environments. Groups of parameters and ablation studies are presented for further analyzing components.

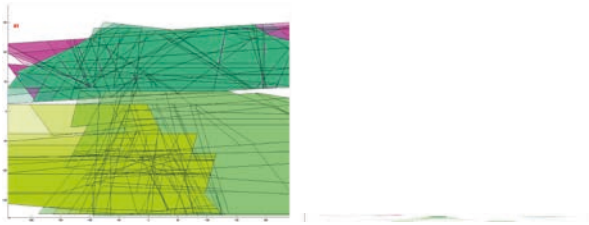

Figure 9. Ablation study for combination for bending surfaces. Compared to the understanding by geometric constraints (right column), it is difficult to estimate bending surfaces without the constraints (left column).

Extracting lines is the cornerstone of understanding a bending path scene. Lines are extracted by algorithms [40]. The parameter $\lambda_l \in [0,1]$ conducts the number of extracted lines. $\lambda_l = 0$ means no extracted lines and $\lambda_l = 1$ is to extract full lines. Therefore an experiment was performed as shown in Figure 8. As to a single capture, lines were extracted via growing λ_l. It indicates that the proposed algorithm is robust to unstable extracted lines. Obviously, when there are no lines, it is unable to understand a scene. As increasing lines are extracted, more bending surfaces are estimated, helping a robot understand more details of bending path scenes.

A bending surface consists of two rectangular planes satisfying the integrity constraints (Eq. 28 and Eq. 29) and orientation constraints (Eq. 30). Also, ablation studies for these two geometric constraints were found as shown in Figure 9. Without the integrity and orientation constraints, it is hard to infer an unstructured bending path.

4.4 Comparison

As to bending bridge scenes from the FDWW dataset [8], experimental comparisons performed between our method and the other two approches [8, 32], with results as shown in Figure 10. Previous approaches approximate a scene using only spatial rectangles [8] and spatial right angles [32]. They can only account for surfaces of those vertical walls (e.g., red, blue and yellow areas) and horizontal floor (e.g., brown areas). But in a field environment, those pathways are always of bending surfaces. Therefore, they are prone to failure in understanding bending pathways. However, by contrast, our method can successfully understand diverse surfaces including not only side surfaces but also bending surfaces, such as turquoise and mint areas, and reconstruct them in a 3D scene.

Our method can address those bending surfaces in an environment. Since there are many bending structures such as bending windscreens of a car in street scenes, then more experiments were performed on Cityscapes dataset [42], which is compared to deep learning-based methods [46], with results as shown in Figure 11. Zhang's framework adopted deep neural networks, but it has difficulty in understanding bending and sloping surfaces such as windscreens of vehicles in a street environment. Our algorithm obtain better understanding of 3D reconstruction of bending sloped scenarios, which helps to improve robot's ability to autonomous navigation.

Besides, experimental comparisons were compared to more deep learning based methods on LSUN dataset [47], with results as shown in Figure 12. Through a fully convolutional neural network, the approach [48] infers only rough layouts, without understanding semantic information such as Manhattan and bending surfaces. However, our algorithm has advantages of interpretability of bending surfaces without any prior training, which is more practical and reliable in navigation.

Experimental comparisons for components were discussed. As shown in Figure 13, the method considers on spatial right angle projections following left or right walls. As a result, it has difficulty estimating bending pathways in field environments with few structured walls. However, the proposed approach is able to account for angle projections of bending surfaces, which helps to understand field bending pathways.

Furthermore, more experimental results were performed on field pathways of different color and illumination. As shown in Figure 14, because the proposed method adopts geometric inference, the approach can successfully understand field pathways, which is robust against changes in illumination and color.

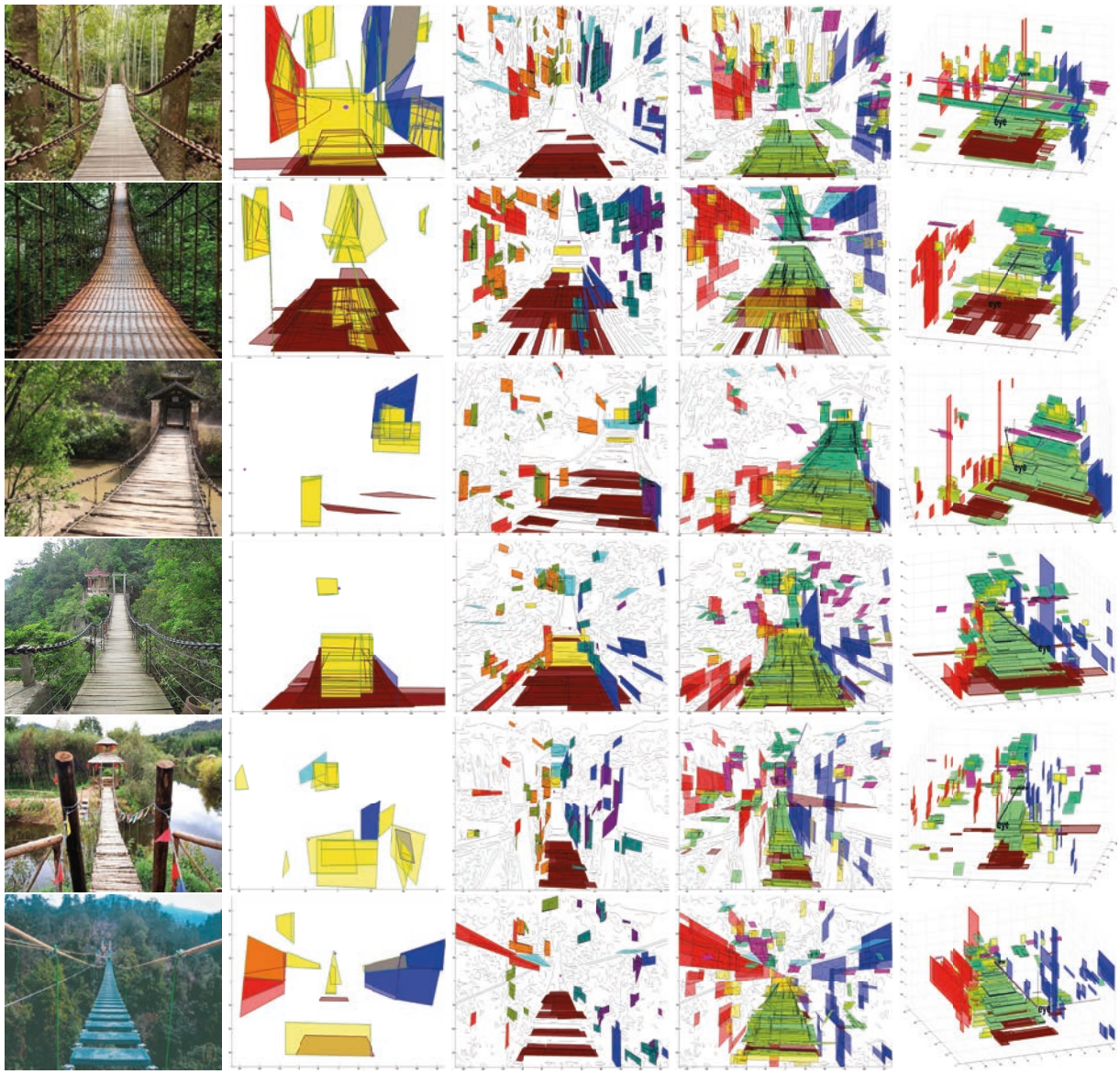

Figure 10. Experimental comparisons. **First** column: bending pathways [8]. **Second** column: Wei's method [8]. **Third** column: Wang's method [32]. **Fourth** column: our understanding. **Right** column: reconstruction. For field bending pathways, the other two methods (first and second column) just focus on spatial surfaces on those vertical walls (e.g., red, blue and yellow areas) and horizontal floor (e.g., brown areas), so they fail to function in representation for bending surfaces in such unstructured environments. By contrast, our method can successfully understand both side structures and bending pathways (e.g., turquoise and mint areas) in a 3D environment.

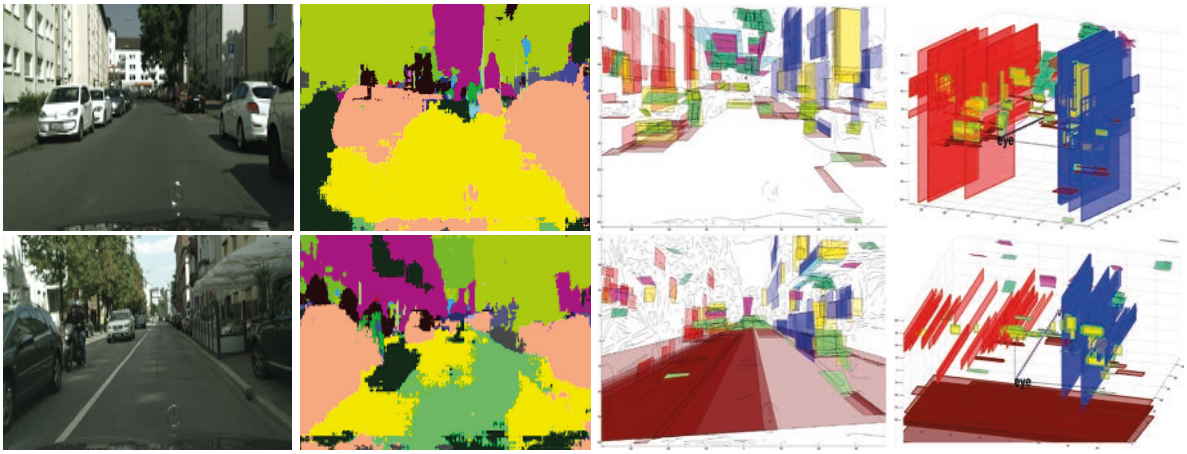

Figure 11. Experimental comparisons. First column: street scenes from Cityscapes dataset [42]. Second column: Zhang's method [46]. Third column: our understanding. Right column: our reconstruction. No satisfactory account of the mechanism that caused the understanding of bending sloped scenarios (e.g., windscreens) was given in Zhang's data-driven algorithm. However, the proposed method has a better understanding of such bending surfaces.

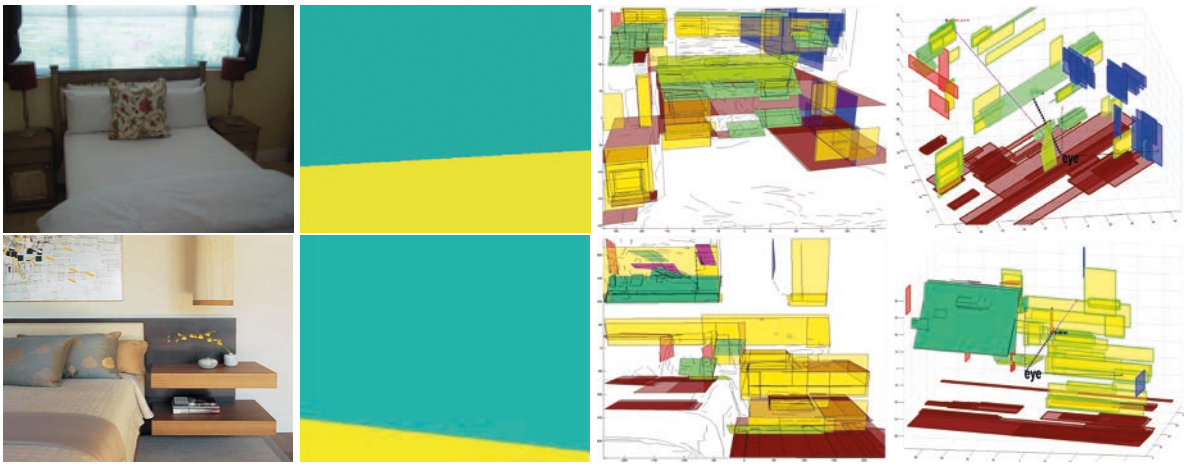

Figure 12. Experimental comparisons. First column: input images [47]. Second column: Lee's method [48]. Third column: our scene understanding. Right column: our reconstruction. Lee's method only gives rough layouts. Based on interpretable geometric inferences, our approach can understand a scene including Manhattan and bending sloped surfaces without prior training.

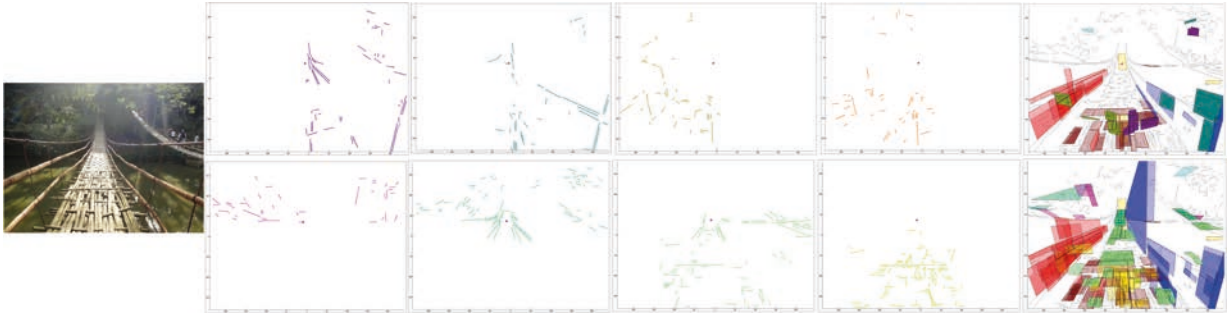

Figure 13. Experimental comparisons for components. Left: input pathway. Top row: spatial right-angles [32]. Bottom row: bending angle projections. Right column: corresponding understanding. The proposed method has advantages in understanding bending surfaces.

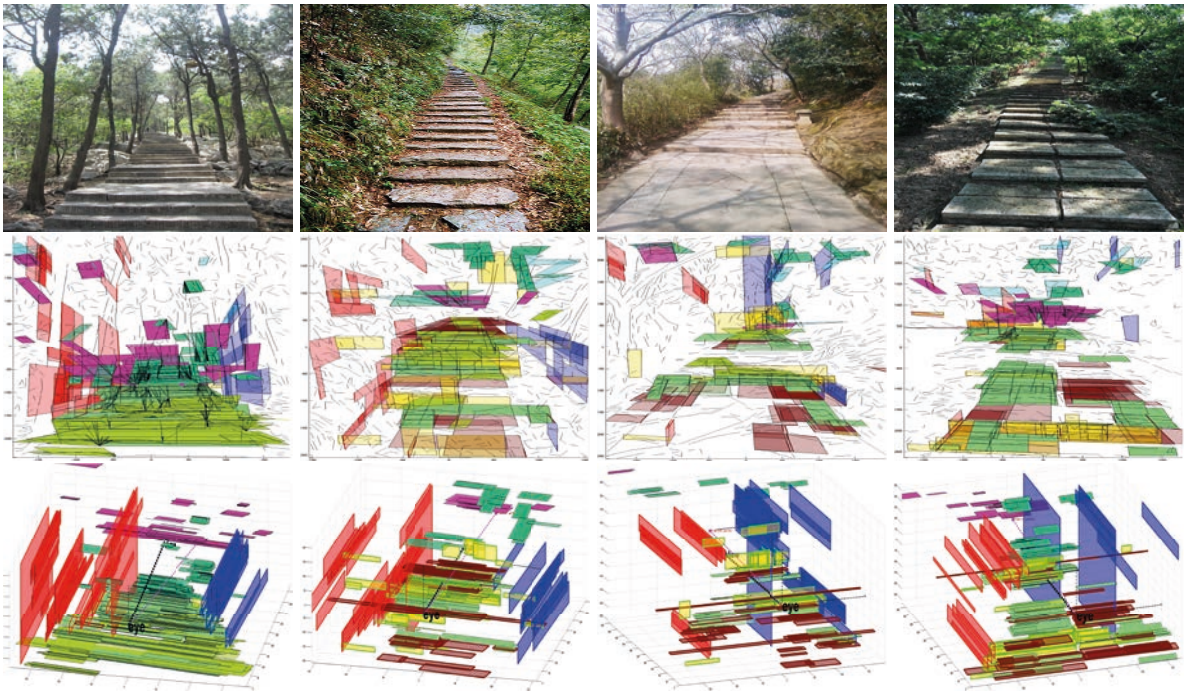

Figure 14. More experimental results. First row: diverse field pathways. Second row: understanding. Third row: reconstruction. For diverse field environments with varying illumination and color, the proposed method is able to understand and reconstruct bending pathways.

5 Conclusions

We presented an algorithm to understand bending paths in unstructured environments from monocular vision without prior training. New clusters of bending angle projections are efficiently extracted and clustered via explainable geometric constraints. Through sub-cluster-pairs, potential bending surfaces are estimated. With combinations that satisfy geometric constraints that are related to integrity and *LVP*, valid bending surfaces are obtained. A bending path scene can be approximated including bending surfaces in 3D reconstruction, and the estimated sloping gradient of bending surfaces can help robot to adjust its speed to climb the bending sloped surfaces. Unlike deep learning-based methods, the proposed algorithm requires no prior training. Based on interpretable geometric constraints, the approach is robust to changes in diverse natural scenes including variation of illumination and color, which is more practical and reliable for mobile robots to autonomously navigate in a complicated natural scene at day or night. Without using precise depth data (e.g., 3D point clouds or RGB-D data), the algorithm has the advantages of lower investment and energy consumption. Estimated bending surfaces in natural scenes were compared to the ground truth through measuring the percentage of pixels that were incorrectly classified. With advantages of interpretability of bending structures, no prior training, and low cost, the proposed system of explainable process and high reliability has broad application prospects, meeting the requirements of autonomous navigation.

Acknowledgments

This work was supported by the NSFC Project (Nos. 62003212, 61771146 and 61375122).

References

[1] E. J. GIBSON and R. D. WALK. The visual cliff. Sci. Am, 202: 64–71, 1960.

[2] Z. J. He and K. Nakayama. Visual attention to surfaces in three-dimensional space. Proc. Natl. Acad. Sci. U. S. A, 92 (24): 11155–11159, 1995.

[3] J. J. Koenderink, A. J. Van Doorn, and A. M. Kappers. Pictorial surface attitude and local depth comparisons. Percept. Psychophys, 58 (2): 163–173, 1996.

[4] Ludovic Magerand and Alessio Del Bue. Revisiting projective structure from motion: A robust and efficient incremental solution. IEEE Trans. Pattern Anal. Mach. Intell., 42(2): 430–443, 2020.

[5] Berta Bescós, Cesar Cadena, and José Neira. Empty cities: A dynamic-object-invariant space for visual SLAM. IEEE Trans. Robotics, 37 (2): 433–451, 2021.

[6] Saifullahi Aminu Bello, Cheng Wang, Naftaly Muriuki Wambugu, and Jibril Muhammad Adam. Ffpointnet: Local and global fused feature for 3d point clouds analysis. Neurocomputing, 461: 55–62, 2021.

[7] Andrea Cavagna, Stefania Melillo, Leonardo Parisi, and Federico Ricci-Tersenghi. Sparta tracking across occlusions via partitioning of 3d clouds of points. IEEE Transactions on Pattern Analysis and Machine Intelligence, 43 (4): 1394–1403, 2021.

[8] Hui Wei and Luping Wang. Understanding of indoor scenes based on projection of spatial rectangles. Pattern Recognition, 81: 497–514, 2018.

[9] Armon Shariati, Bernd Pfrommer, and Camillo J. Taylor. Simultaneous localization and layout model selection in manhattan worlds. IEEE Robotics and Automation Letters, 4(2): 950–957, 2019.

[10] Luping Wang and Hui Wei. Indoor scene understanding based on manhattan and non-manhattan projection of spatial right-angles. J. Vis. Commun. Image Represent., 80: 103307, 2021.

[11] Jeong-Kyun Lee and Kuk-Jin Yoon. Joint estimation of camera orientation and vanishing points from an image sequence in a non-manhattan world. International Journal of Computer Vision, 127 (10): 1426–1442, 2019.

[12] Ifham Abdul Latheef Ahmed and Mohamed Hisham Jaward. Classifier aided training for semantic segmentation. Journal of Visual Communication and Image Representation, 78: 103177, 2021.

[13] Sudhanshu Mittal, Maxim Tatarchenko, and Thomas Brox. Semi-supervised semantic segmentation with high- and low-level consistency. IEEE Transactions on Pattern Analysis and Machine Intelligence, 43 (4): 1369–1379, 2021.

[14] Petra Bosilj, Erchan Aptoula, Tom Duckett, and Grzegorz Cielniak. Transfer learning between crop types for semantic segmentation of crops versus weeds in precision agriculture. Journal of Field Robotics, 37 (1): 7–19, 2020.

[15] Lucas M. Tassis, Joao E. Tozzi de Souza, and Renato A. Krohling. A deep learning approach combining instance and semantic segmentation to identify diseases and pests of coffee leaves from in-field images. Computers and Electronics in Agriculture, 186: 106191, 2021.

[16] Andreas Bar, Jonas Lohdefink, Nikhil Kapoor, Serin John Varghese, Fabian Huger, Peter Schlicht, and Tim Fingscheidt. The vulnerability of semantic segmentation networks to adversarial attacks in autonomous driving: Enhancing extensive environment sensing. IEEE Signal Processing Magazine, 38 (1): 42–52, 2021.

[17] Luping Wang and Hui Wei. Recognizing slanted deck scenes by non-manhattan spatial right angle projection. IEEE Intelligent Systems, 37 (5): 75–85, 2022.

[18] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo. Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. IEEE Transactions on Intelligent Transportation Systems, 19 (1): 263–272, 2018.

[19] F. Lateef, M. Kas, and Y. Ruichek. Saliency heat-map as visual attention for autonomous driving using generative adversarial network (gan). IEEE Transactions on Intelligent Transportation Systems, pages 1–14, 2021.

[20] Luping Wang and Hui Wei. Understanding of curved corridor scenes based on projection of spatial right-angles. IEEE Transactions on Image Processing, 29: 9345–9359, 2020.

[21] Leonardo Cabrera Lo Bianco, Jorge Beltrn, Gerardo Fernndez Lpez, Fernando Garca, and Abdulla Al-Kaff. Joint semantic segmentation of road objects and lanes using convolutional neural networks. Robotics and Autonomous Systems, 133: 103623, 2020.

[22] Ivan Kreso, Josip Krapac, and Sinisa Segvic. Efficient ladder-style densenets for semantic segmentation of large images. IEEE Transactions on Intelligent Transportation Systems, 22 (8): 4951–4961, 2021.

[23] Gjorgji Nikolovski, Michael Reke, Ingo Elsen, and Stefan Schiffer. Machine learning based 3d object detection for navigation in unstructured environments. In IEEE Intelligent Vehicles Symposium Workshops (IV Workshops), pages 236–242, 2021.

[24] Galadrielle Humblot-Renaux, Letizia Marchegiani, Thomas B. Moeslund, and Rikke Gade. Navigation-oriented scene understanding for robotic autonomy: Learning to segment driveability in egocentric images. IEEE Robotics and Automation Letters, 7 (2): 2913–2920, 2022.

[25] Andreas Kloukiniotis and Konstantinos Moustakas. Vanishing point detection based on the fusion of lidar and image data. In 30th Mediterranean Conference on Control and Automation, MED 2022, Vouliagmeni, Greece, June 28 - July 1, 2022, pages 688–692. IEEE, 2022.

[26] Marin Orsic and Sinisa Segvic. Efficient semantic segmentation with pyramidal fusion. Pattern Recognition, 110: 107611, 2021.

[27] Christopher J. Holder and Toby P. Breckon. Learning to drive: End-to-end off-road path prediction. IEEE Intell. Transp. Syst. Mag., 13 (2): 217–221, 2021.

[28] Bhakti Baheti, Shubham Innani, Suhas S. Gajre, and Sanjay N. Talbar. Semantic scene segmentation in unstructured environment with modified deeplabv3+. Pattern Recognit. Lett., 138: 223–229, 2020.

[29] Kasi Viswanath, Kartikeya Singh, P. Jiang, P. B. Sujit, and S.Saripalli. OFFSEG: A semantic segmentation framework for off-road driving. In 17th IEEE International Conference on Automation Science and Engineering, CASE , Lyon, France, pages 354–359, 2021.

[30] Luping Wang and Hui Wei. Avoiding non-manhattan obstacles based on projection of spatial corners in indoor environment. IEEE/CAA Journal of Automatica Sinica, 7: 1190 – 1200, 2020.

[31] Hui Wei and Luping Wang. Visual navigation using projection of spatial right-angle in indoor environment. IEEE Transactions on Image Processing, 27(7): 3164–3177, 2018.

[32] Luping Wang and Hui Wei. Curved alleyway understanding based on monocular vision in street scenes. IEEE Transactions on Intelligent Transportation Systems, 23 (7): 8544–8563, 2022.

[33] Paolo Arena, Carmelo Fabrizio Blanco, Alessia Li Noce, Salvatore Taffara, and Luca Patane. Learning traversability map of different robotic platforms for unstructured terrains path planning. In International Joint Conference on Neural Networks (IJCNN), pages 1–8, 2020.

[34] Paolo Arena, Fabio Di Pietro, Alessia Li Noce, Salvatore Taffara, and Luca Patanè. Assessment of navigation capabilities of mini cheetah robot for monitoring of landslide terrains. In 6th IEEE International Forum on Research and Technology for Society and Industry, RTSI 2021, Naples, Italy, September 6-9, 2021, pages 540–545. IEEE, 2021.

[35] Luping Wang and Hui Wei. Understanding of wheelchair ramp scenes for disabled people with visual impairments. Engineering Applications of Artificial Intelligence, 90: 103569, 2020.

[36] David D. Fan, Ali-akbar Agha-mohammadi, and Evangelos A. Theodorou. Learning risk-aware costmaps for traversability in challenging environments. IEEE Robotics and Automation Letters, 7 (1): 279–286, 2022.

[37] Wenbo Dong, Pravakar Roy, and Volkan Isler. Semantic mapping for orchard environments by merging two-sides reconstructions of tree rows. Journal of Field Robotics, 37 (1): 97–121, 2020.

[38] Daniel Maturana, Po-Wei Chou, Masashi Uenoyama, and Sebastian A. Scherer. Real-time semantic mapping for autonomous off-road navigation. In Field and Service Robotics, Results of the 11th International Conference, FSR , Zurich, Switzerland, volume 5, pages 335–350, 2017.

[39] Yi Yang, Di Tang, Dongsheng Wang, Wenjie Song, Junbo Wang, and Mengyin Fu. Multi-camera visual SLAM for off-road navigation. Robotics Auton. Syst., 128: 103505, 2020.

[40] Fowlkes C Arbelaez P, Maire M. From contours to regions: An empirical evaluation. In CVPR, pages 2294–2301, 2009.

[41] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. Int. J. Robotics Res., 32 (11): 1231–1237, 2013.

[42] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3213–3223, 2016.

[43] Kai A. Metzger, Peter Mortimer, and Hans-Joachim Wuensche. A fine-grained dataset and its efficient semantic segmentation for unstructured driving scenarios. In 25th International Conference on Pattern Recognition, ICPR , Virtual Event / Milan, Italy, pages 7892–7899, 2020.

[44] Maggie B. Wigness, Sungmin Eum, John G. Rogers, David Han, and Heesung Kwon. A RUGD dataset for autonomous navigation and visual perception in unstructured outdoor environments. In IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS , Macau, SAR, China, pages 5000–5007, 2019.

[45] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. CVPR, pages 3485 – 3492, 2010.

[46] Y. Zhang, P. David, H. Foroosh, and B. Gong. A curriculum domain adaptation approach to the semantic segmentation of urban scenes. IEEE Transactions on Pattern Analysis and Machine Intelligence, 42 (8): 1823–1841, 2020.

[47] Y. Zhang, F. Yu, S. Song, P. Xu, A. Seff, and J. Xiao. Largescale scene understanding challenge: Room layout estimation. 2016.

[48] Chen-Yu Lee, Vijay Badrinarayanan, Tomasz Malisiewicz, and Andrew Rabinovich. Roomnet: End-to-end room layout estimation. In IEEE International Conference on Computer Vision, ICCV, pages 4875–4884, 2017.

Luping Wang received the Ph.D. degree at the department of computer science at Fudan University in 2019. Since August 2019, he has joined the department of electrical engineering at University of Shanghai for Science and Technology. His current research interests include scene understanding, visual navigation, artificial intelligence and cognitive science.
https://orcid.org/0000-0002-2417-4561

Hui Wei received the Ph.D. degree at the department of computer science at Beijing University of Aeronautics and Astronautics in 1998. From 1998 to 2000, he was a postdoctoral fellow at the department of computer science and the institute of artificial intelligence at Zhejiang University. Since November 2000, he has joined the department of computer science and engineering at Fudan University. His research interests include artificial intelligence and cognitive science.
https://orcid.org/0000-0003-2696-0707