Effects of Ground Manifold Modelling on the Accuracy of Stixel Calculations

Noor Haitham Saleem, Hsiang-Jen Chien, Mahdi Rezaei, and Reinhard Klette

***Abstract*—This paper highlights the role of ground manifold modelling for stixel calculations; stixels are medium-level data representations used for the development of computer vision modules for self-driving cars. By using single disparity maps and simplifying ground manifold models, calculated stixels may suffer from noise, inconsistency, and false-detection rates for obstacles, especially in challenging datasets. Stixel calculations can be improved with respect to accuracy and robustness by using more adaptive ground manifold approximations. A comparative study of stixel results, obtained for different ground-manifold models (e.g. plane-fitting, line-fitting in v-disparities or polynomial approximation, and graph cut), defines the main part of this paper. The paper also considers the use of trinocular stereo vision and shows that this provides options to enhance stixel results compared to binocular recording. Comprehensive experiments are performed on two publicly available challenging datasets. We also use a novel way for comparing calculated stixels with ground truth. We compare depth information, as given by extracted stixels, with ground-truth depth, provided by depth measurements using a highly accurate LiDAR range sensor (as available in one of the public datasets). We evaluate the accuracy of four different ground-manifold methods. Experimental results also include quantitative evaluations of the trade-off between accuracy and run time. As a result, the proposed trinocular recording together with graph-cut estimation of ground manifolds appears to be a recommended way, also considering challenging weather and lighting conditions.**

***Index Terms*—Ground manifold, v-disparity, stixels, monocular, binocular, trinocular, membership function, obstacle height, dynamic programming**

I. Introduction

STIXELS are "stick elements". They have been introduced in computer graphics in [1], and defined recently a useful way for describing 3-dimensional (3D) scenes in computer vision [2], especially in the context of *vision-based driver-assistance systems* (VB-DAS).

VB-DAS are integral components of modern cars [3]. Besides cameras, other types of sensors are also commonly used, defining the more generic *advanced driver assistance systems* (ADAS), being a development towards autonomous vehicles. The designed systems aim at an understanding of traffic environments in order to improve traffic safety and efficiency [4], and also for better travel comfort. Examples of ADAS technologies are auto-braking systems, evasive steering assistance, or blind spot monitoring.

We briefly define three basic terms used in this paper. The *ground manifold* is the estimated surface function for road and adjacent levelled areas; a plane defines the simplest model (i.e. a *ground plane* [5]); in this paper we consider different surface functions as models for the ground manifold. The *ego-vehicle* is the vehicle in which the system is operating in [6]. The *free space* is a region ahead of the ego-vehicle where this vehicle may potentially (i.e. safely) drive in, for example, in the next few seconds [7], [8].

In 2009, stixels have been proposed as a *medium-level* (i.e. between pixel data and semantic segments) representation for urban road scenes. This compact representation of disparity maps aims at simplifying subsequent semantic segmentation of a given scene. A projectively recorded scene can be mapped into a top-down view, to be divided into adjacent cells of an *occupancy grid*. Cells of this grid are of size $w \times w$ measured in pixels. Disparities, measured for real-world objects within one cell of this grid, are assumed to be about at the same depth. A stixel [6] forms now a vertical "stick" above such a $w \times w$ base cell; in this original definition it is a square-base thin column on a ground plane (i.e. on a regular occupancy grid) as shown in Fig. 1.

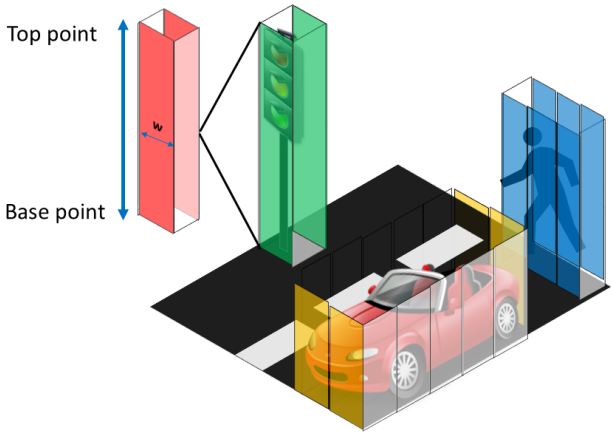

Fig. 1. Stixels (vertical sticks) describing obstacles: The (original) stixel has a square base, and goes from a defined ground plane to the top of an object, located on the stixel's square base.

A stixel maps pixels that belong to an object (i.e. which are at about the same distance to the recording camera) vertically into "columns" [2], sitting on the ground plane. A stixel is ideally upper-bounded by the top of an object. See Fig. 2 for such a representation in a real-world scene. Technical terms used in the caption of this figure (e.g. "cost image") are explained later; this figure indicates at this point a general process of stixel calculation defined by disparity-map calculation (top-left), base-point detection in the ground

N. H. Saleem, H.-J. Chien, M. Rezaei, and R. Klette are with the Electrical and Electronic Engineering Department, School of Engineering, Computer and Mathematical Sciences, Auckland University of Technology, Auckland, New Zealand. Email: {*nalani, jchien, mahdi.rezaei, rklette*}*@aut.ac.nz.*

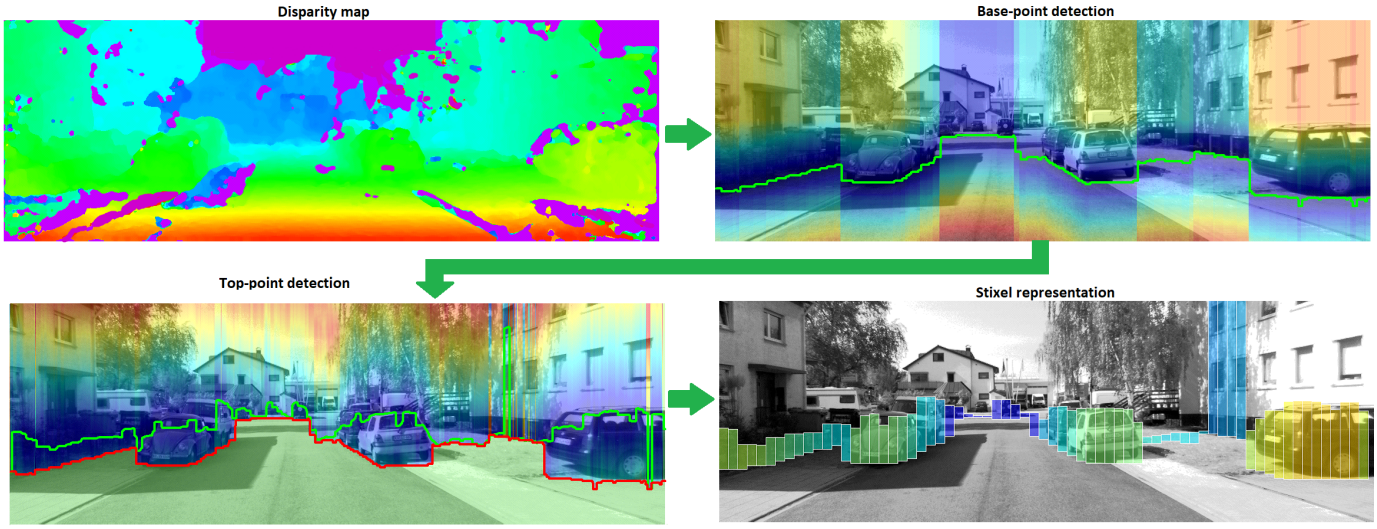

Fig. 2. Stixel representation of a street scene. *Top-left*: Disparity map (using an SGM-variant for stereo matching) visualized by applying a color key. *Top-right*: Base-point selection by a minimum cut (shown in green) through a cost image. Regions in deep blue show lower costs which are preferred by a dynamic programming-based optimizer. *Bottom-left*: Top-point selection by a minimum-cut (shown in green) through a cost image, subject to the base points (shown in red). *Bottom-right*: Extracted stixels.

manifold (top-right), followed by top-point detection (bottom-left), and the resulting stixels (bottom-right). Generated stixels, also called the stixel world, surround the free space in the assumed ground plane.

Ground-plane (or ground-manifold) estimation can be approached using either monocular or multi-ocular vision [10]. Monocular vision also supports ways of distance estimation (e.g. by inverse perspective mapping); see [11]. There are also combined monocular-binocular stixel methods; free-space is estimated by using a single camera only, followed by obstacle detection using stereo vision [36]. In order to detect free-space from a single camera, we may employ a time-efficient lane-based free-space detection method [8]. For example, lane detection can be performed by using a Hough transform for straight lines following edge detection; the Hough transform is a basic method for line extraction [51].

Figure 3 illustrates possible steps: Cropping of a recorded frame into a defined *region of interest* (ROI), edge detection using the Sobel operator due to its "unbiased" definition, and straight line detection by application of an optimised Hough transform; the transform is applied recursively, using optimized (Otsu algorithm [52]) threshold values, until a predefined number of lines is found, or the threshold reaches its minimum. Finally, that "dominant" pair of lines with the best correspondence in angular directions is selected for specifying road contours (i.e. the free-space) in such a monocular vision approach.

As illustrated by Fig. 3, there remain many spaces which were not properly estimated regarding free-space or possible base-points of obstacles; these deficiencies would yield an early estimation of obstacles.

Robust obstacle segmentation and scene understanding are key tasks for visual sensors (cameras) in self-driving cars for being able to interpret dynamic environments. Cameras are playing a significant role in autonomous driving; they are capable of providing rich information including distances to obstacles given in traffic scenes.

Currently emerging vehicle testbeds (e.g. equipped with sensors along roads, and vehicle-to-infrastructure communication; see [21] for an example) aim at exact and comparative evaluations of control components designed for driver assistance or driver-less vehicles. Having different options for sensors and ground-manifold models, it is, of course, important to compare efficiencies and possible accuracies of stixel calculations. Accuracy of stixels requires a disparity signal of "good" quality; this quality often decreases in cases of occlusions or textureless image patches [22]. Since noisy 3D points have a considerable impact on ground-manifold estimation, it is crucial to identify unreliable disparity values before they are transformed into 3D space and used for stixel estimation. Unfortunately, these issues are common in traffic scenes, thus more efforts are needed to improve disparity signals, also aiming at more reliable free-space estimation and stixel calculations. Due to road-geometry variations, and difficulties in recording those properly (e.g. due to weather conditions or traffic density), there is ongoing work to improve

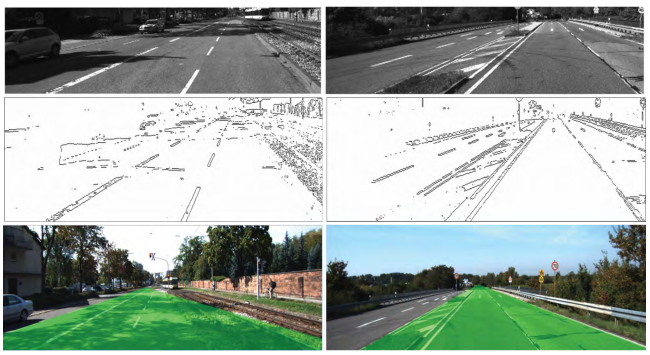

Fig. 3. Free-space detection using monocular vision, shown for two images of the KITTI *road* dataset. *Top*: Selected ROI (i.e. "middle rows" of a frame only). *Middle*: Edge detection using Sobel. *Bottom*: Detected free-space.

ground-manifold estimation and stixel calculation. However, current research as reported in [4], [23], [24] still uses just a ground-plane for modelling the road-surface; we discuss how this is prone to errors as road-geometry is not always perfectly planar.

The reminder of this paper is structured as follows: Section II provides at first a brief literature review on work related to stixel computation. Section III recalls then previously specified ways of stixel construction, considering the used ground-manifold specification as a variable subprocess. In Section IV, a number of methods is deployed to detect a ground manifold for stixel construction. Section V explains how to use trinocular recording for stixel calculations. Section VI evaluates effects of those different ground-manifold models, and of bi- or trinocular recording. Section VII concludes.

II. Related Work

We briefly discuss work on road surface and stixel extraction, which are both considered to be crucial steps towards stixel calculation. Road boundary segmentations are applicable for modelling the space where an ego-vehicle is potentially driving in. Detected road boundaries support concepts of ground versus obstacle segmentation.

Stixel calculation requires depth estimates and is better approached by multi-ocular stereo vision or other depth sensors; thus it also makes sense to use depth data also already for the ground-manifold estimation step. Vision sensors and related data analysis define a core component in ADAS; A binocular vision system depends on calculated disparity values for calculating scene depth; disparities are calculated by implementing stereo matching algorithms [16] on images obtained by a left and right camera. There might be pre-processing applied before the stereo matching step, such as in [17], [18], for enhancing matching outcomes. Results can be filtered by applying confidence measures; see [19] for various stereo-matching confidence measures. Stereo-vision results may also be improved by using a trinocular vision system rather than just a binocular one; see, for example, [20].

A row-wise histogram of a calculated *disparity map* D is known as *v-disparity map* [5], where v denotes row coordinates of an image. The analysis of v-disparity maps (e.g. calculations of lower envelops, or other forms of curve approximations) defines a common way for ground-manifold estimation. Noise in disparity maps results in noise in v-disparity maps. It is challenging to identify an "ideal" curve in v-disparity space using a curve-approximation method and v-disparity for binocular vision alone. Stereo vision supports the use of techniques such as v-disparity representation [5], disparity analysis [31], or occupancy grid generation [2], [32].

Rapid stixel-based analysis enhances stixel extraction by having lower computational costs; in [35] a direct stixel computation is presented by changing the parametrization from disparity space into a pixel-wise cost volume for speed improvement. In [36], the authors use deep convolutional neural networks for free-space detection using monocular vision, while obstacle detection and stixel calculation is done by using stereo vision. A fast stixel computation without using depth maps is proposed in [37]. It supports high-speed pedestrian detection (at the speed of 200 fps).

Color fusion models compute stixels by using stereo images (i.e. depth cues) in combination with color appearance. Such methods have been presented for stixel segmentation [22], [38], [39]; their implementation can be done by using a low-level fusion of depth with image signals or semantic information in the stixel generation process. Scharwächter et al. employed pixel classification with random decision forests [38], while in [39] semantic information via object detectors is used for a suitable set of classes. Yet another method has been presented in [22] to improve stixels using low-level appearance models in an on-line self-supervised framework. Recently, joint stixel representations, combining semantic data and depth, are proposed to integrate both categories in terms of a joint optimized scene model [25].

Despite the proven effectiveness, such techniques may also have negative impacts on stixel segmentation [25]. Rapid stixel-based methods have some drawbacks which are prone to low depth accuracy, which in turn affects stixel extraction negatively. Therefore, we consider the use of stereo-matching confidence maps (see [19] for different options for such maps) with the aim of improving stixel segmentation. (Effects of confidence-involvements contributed to the images shown in Fig. 2.) We focus on a careful analysis for identifying a recommended way for curve detection (i.e. ground-manifold estimation) in v-disparity space. With promising results achieved by employing optimization techniques, this paper provides

- a new method, called *trinocular graph-cut*, for generating a robust lower envelope in v-disparity space to improve stixel detection, verified on KITTI data,
- a new ground-truth measure for stixel accuracy evaluation, proposed for the 6D Vision Dataset, and
- an extensive analysis of a low-cost and accurate architecture for reducing false-positives in stixel estimation using a model with a reduced number of parameters for ground-manifold detection.

III. Stixel Representation

A stixel starts on top with a detected upper "end" of an object and ends at the bottom on the ground plane (or ground manifold in general, also addressing non-planar surfaces). Stixels are computed from a disparity map[1] D at three stages:

1) *Base point detection.* Base points are identified by locating the boundary of free space in the given image. The boundary is found by first building an occupancy map from range data above an estimated road manifold, then solving for an optimal cut separating free space from the rest of the grid cells in the map.
2) *Height segmentation.* Foreground pixels are separated from the background, and an upper boundary (i.e. top points) of obstacles "resting on the ground" are detected.
3) *Stixel extraction.* Column-wise obstacles are grouped and represented by bounding boxes, and depth values

[1]We adopt a *semi-global matching* (SGM) algorithm [16] for disparity calculation.

of pixels in the same group are integrated to form a stixel with one unified depth value.

In this section we provide an in-depth walk-through for this process following papers which introduced stixels as a medium-level scene representation.

A. Base Point Detection

The first step of stixel construction is to find the bottom of the closest obstacle for every column [2], [32]. The search is based on the the free space analysis by means of *occupancy grids* [40], which represent the scene as a 2D discrete map. The use of occupancy grids for free space detection dates back to 2007 [7]. A probabilistic occupancy grid can be built by projecting depth data (or, equivalently, disparities) along the Y-axis of the camera (this axis goes from ground plane upward) into the ground plane, and then by *binning* the projected data using a 2D histogram. The grid can be defined in either 2D Cartesian coordinates (on the XZ ground plane) or in polar coordinates. In the latter case, the grid shows a distribution of pixels in the column-disparity space, which is also known as a u-disparity map [2] (contrary to the v-disparity map that is introduced in the following section). An example is shown in Fig. 4.

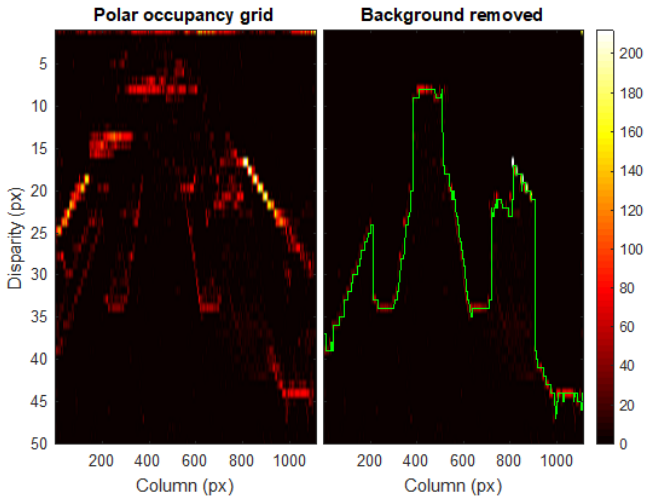

Fig. 4. Occupancy map showing the distribution of objects above the road surface. *Left:* Computed polar-occupancy grid. *Right:* After background object removal. The green curve visualises a column-by-column maximum cut found by means of dynamic programming. The larger a disparity, the closer is the object to the camera.

To correctly find the free space from an occupancy map, the ground manifold has to be estimated to include only obstacles above the ground to build the grid. Details regarding the estimation of ground manifold are discussed in Section IV.

By means of an occupancy grid, the free space is efficiently found using a graph-cut algorithm. The nearest prominent object is first identified for each column, and the grid cells behind are occluded. After removing background objects from the occupancy map, a dynamic programming technique is carried out to locate the maximum-cut through the map that separates free space and the obstacles [7]. For each column x in disparity map D, the process decides a disparity d, as illustrated in Fig. 4. Back-projecting such a cut in the occupancy map to the image-disparity space and subsequently into the Euclidean space defines an obstacle manifold, as rendered in red in Fig. 5.

[2]In the original paper (u, v) is used to denote image coordinates; we are using (x, y) for image coordinates.

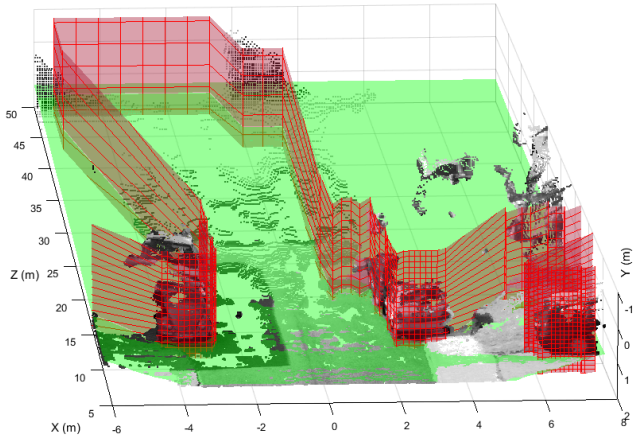

Fig. 5. Reconstructed 3D points from a disparity image, road manifold (green), and obstacle manifold (red) from an occupancy map.

At the end of this stage, a base point is decided for each column of D by locating the intersection of the obstacle manifold and road manifold (see Fig. 5). The per-pixel distances between the road manifold and obstacle manifold are then computed as a cost function for deciding base points (see Fig. 2 for example). The minimum cut through the cost then defines the base points of stixels, as represented as a set of row indices $\{b_1, b_2, \ldots, b_{N_{\text{col}}}\}$ where N_{col} is the number of columns of the image domain, and (x, b_x) denotes the image coordinates of base point in column x.

B. Height Segmentation

The height of obstacles, which sit on the ground manifold, is obtained by seeking an ideal segmentation between *foreground* and *background* disparities. The goal of the stage is to find top points $t_1, t_2, \ldots, t_{N_{\text{col}}}$ that together with those base points, that are found at the previous stage, define the span of obstacles in a column-wise manner.

In [6], the height-of-obstacle calculation begins with selecting membership votes. Briefly, the membership values rely on the selection of every disparity of each column from the disparity for its member to the foreground obstacle. A membership value can be positive if it does not exceed the maximum distance of the expected obstacle disparity; otherwise, it will be negative.

The Boolean membership vote brings the challenge to identify a threshold value for the distance; if this value is too large then all disparities will be chosen from the foreground membership, and vice-versa. Therefore, the application of Boolean membership in a continuous variation is a better

alternative with an exponential function of the form

$$M(x,y) = \begin{cases} 2^{1-\delta^2(x,y)} - 1, & \text{if } y < b_x \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where

$$\delta(x,y) = \frac{d_x - D(x,y)}{d_x - Z^{-1}\left(Z(d_x) + \triangle Z\right)} \quad (2)$$

with $d_x = D(x, b_x)$, the disparity of an obstacle's base point in column x, Z as the disparity-to-depth conversion function, and $\triangle Z$ as a defined soft constraint range in depth. The approximated Boolean function is illustrated by Fig. 6.

An example of evaluated membership values is shown in Fig. 7. Foreground regions end at the bottom of each contributing column at a base point.

The next step is to decide the boundary between foreground and background votes from the membership function. For this purpose, the cost image is computed as follows:

$$C(x,y) = \begin{cases} \sum_{j=1}^{y-1} M(x,j) - \sum_{j=y}^{b_x} M(x,j), & \text{if } y < b_x \\ \infty & \text{otherwise} \end{cases} \quad (3)$$

A minimum cut, that divides the cost image into upper and lower parts, is then found by using a dynamic programming technique as in [6], while maintaining a smoothness constraint. The cut defines the top points $\{t_1, t_2, \ldots, t_{N_{\text{col}}}\}$. (There are further options for calculating such a cut; we selected due to performance results.)

A visualization of a cost image, used for the height segmentation, was already illustrated in Fig. 2. As can be seen, there are lower costs which show a high likelihood for performing a foreground-background separation.

C. Stixel Extraction

Stixels are extracted by combining at first base points $b_1, b_2, ...b_{N_{\text{col}}}$, obtained as outlined in Section III-A, and top-points $t_1, t_2, ...t_{N_{\text{col}}}$, calculated as per Section III-B; then, a column-wise grouping technique, proposed in [2], [41], is carried out. Given $w \in \mathbb{Z}^+$, a predefined width of stixels, every w neighboring columns are grouped across the whole image, resulting in $\lfloor \frac{N_{\text{col}}}{w} \rfloor$ non-overlapping stixels.

For the i-th stixel we have a set of w base points $B_i = \{b_{x_i}, b_{x_i+1}, ..., b_{x_i+w-1}\}$ and a set of w top points $T_i = \{t_{x_i}, t_{x_i+1}, ..., t_{x_i+w-1}\}$, where $x_i = (i-1)w + 1$. The rectangle spanned from column $x = x_i$ to $x = x_i + w - 1$, and from row $y = \min(T_i)$ to $y = \max(B_i)$, defines the *scope of a stixel* in the image domain.

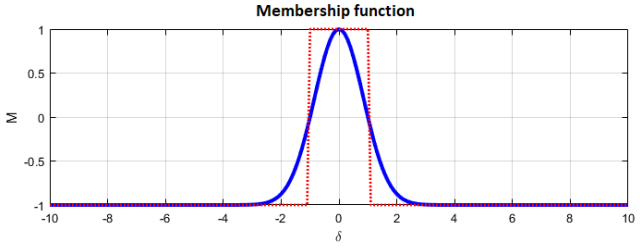

Fig. 6. Exponential membership function (blue) adopted to approximate the Boolean membership (red). The width of the function is determined by ΔZ in (2).

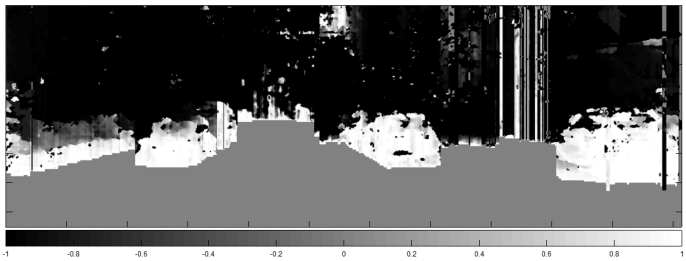

Fig. 7. Evaluated membership of pixels in background (black) and foreground (white) classes. For pixels below the base points, the membership value remains undefined (grey).

Instead of using only base points' disparities, all the disparities within the scope are integrated to yield a more robust estimation of the stixel's depth z_i, by means of a histogram-based regression technique proposed in [2].

Stixel detection represents also a way for ground manifold estimation; all the base points of stixels can act as interpolation points for ground-obstacle segmentation using geometry data with the aim of improving the accuracy. Besides that, a stixel clearly represents the height of the first obstacle facing the vehicle along a given viewing direction. Resulting stixels have been illustrated in Fig. 2, bottom-right. The colours of the stixels encode the distance to the ego-vehicle. Red-scale colours represent closer objects while blue-scale colours represent farther objects.

The accuracy of extracted stixels is directly affected by the estimated ground manifold. In the following section we provide details about ground-manifold estimation methods.

IV. Ground Manifold Modelling

A ground manifold, found at this stage, may be coded as a disparity map G where $G(x,y)$ stores the disparity of the ground at pixel location (x,y). Let D be the disparity map computed by stereo matching, pixel (x,y) is considered to be *above the ground manifold* if $D(x,y) > G(x,y) + \varepsilon$, where $\varepsilon > 0$ defines a tolerance margin.

A variety of methods has been proposed in literature [5], [7], [10], [20], [42] to obtain map G. Some methods directly work on raw data, such as image intensities, disparities, or 3D points, while others apply data projections to reduce the dimensionality of the raw data. Direct methods and projection-based methods are reviewed in this section.

A. Plane Fitting

In a typical road scene, the ground manifold is the dominating surface that lower bounds other objects in the scene. In this case, the manifold can be identified by finding the best-fit 3D surface given to a set of 3D points.

When the ground manifold is assumed to be flat, the estimation can be approached by means of 3D plane fitting. In case that the 3D points are derived from a disparity map, the fitting can be done directly in the image-disparity space. This is shown as follows.

Consider a plane $a_0X + a_1Y + a_2Z + a_3 = 0$ in 3D Euclidean space with plane coefficients $a_0, \ldots, a_3 \in \mathbb{R}$. A point (X, Y, Z) in 3D space is mapped onto an image pixel (x, y) following the pinhole model

$$x = f_x \cdot \frac{X}{Z} + x_c, \quad y = f_y \cdot \frac{Y}{Z} + y_c \tag{4}$$

where (f_x, f_y) are the focal lengths, and (x_c, y_c) is the principal point.

Two calibrated and horizontally rectified pinhole cameras introduce a disparity space, where every pixel (x, y) in the (say) left image is mapped to $(x - d, y)$ in the right image via $d \in [0, d_{\max})$, the disparity value bounded by $d_{\max}$. The disparity-to-depth conversion follows

$$Z = f_x \cdot \frac{B}{d} \tag{5}$$

where B is the length of the baseline (connecting the focal points of the two cameras) in world units.

By first substituting (4) into the plane equation, resulting in

$$a_0 \cdot \frac{Z}{f_x}(x - x_c) + a_1 \cdot \frac{Z}{f_y}(y - y_c) + a_2 Z + a_3 = 0 \tag{6}$$

and then (5) into (6) producing

$$a_0 \cdot \frac{x - x_c}{f_x} + a_1 \cdot \frac{y - y_c}{f_y} + a_2 + a_3 \frac{d}{B f_x} = 0 \tag{7}$$

the plane in the Euclidean space is now modelled in the image-disparity space as another plane:

$$a'_0 x + a'_1 y + a'_2 d + a'_3 = 0 \tag{8}$$

in terms of $a'_0 = (B f_y) a_0$, $a'_1 = (B f_x) a_1$, $a'_2 = f_y a_3$ and $a'_3 = (B f_x f_y) a_2 - (B f_y x_c + B f_x y_c)$. This way the road plane can be found without any need of back-projecting a disparity map into the 3D Euclidean space [43].

An example of a road manifold, modelled in the image-disparity space using the proposed plane fitting technique, is shown in Fig. 8.

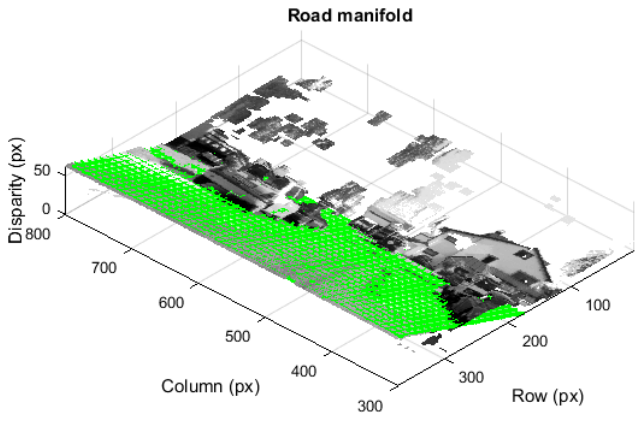

Fig. 8. Road manifold (green) found using the plane-fitting technique in image-disparity space.

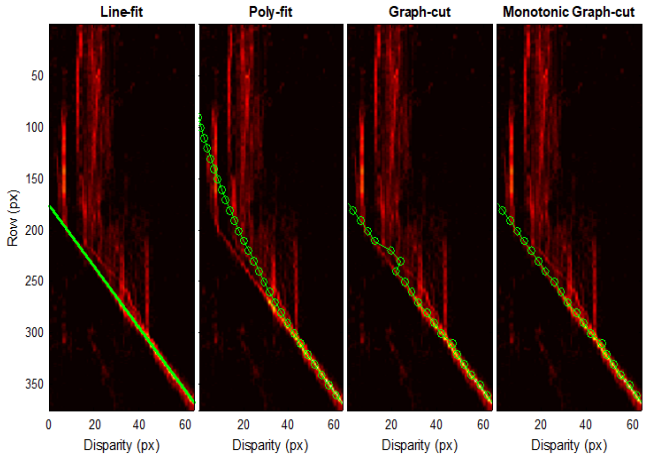

Fig. 9. Demonstration of v-disparity-based ground-manifold modelling. *First column*: Line fitting. *Second column*: Polynomial-based curve fitting. *Third column*: Graph-cut-based curve fitting. *Fourth column*: Graph-cut-based curve fitting with enforced monotonicity.

B. Line Fitting

When the height of the road manifold does not change significantly along the image's x-axis, the plane model in (8) reduces to a line:

$$d = -\frac{a'_1}{a'_2} y - \frac{a'_3}{a'_2} = my + b \tag{9}$$

which turns road manifold estimation into a line-fitting problem of seeking the best-fit line model (m, b).

A computationally efficient way to find the best-fit line is to use a histogram that models the distribution of (y, d) in 2D space. Such a histogram is known as a *v-disparity* or *row-disparity* map [5]. A v-disparity map is computed by accumulating pixels in the same disparity interval in one row y, $1 \le y \le N_{\text{row}}$, of the disparity map:

$$V(y, d) = \text{card}\{x : 1 \le x \le N_{\text{col}} \wedge Q(D(x, y)) = d\} \tag{10}$$

where $0 \le d \le d_{\max}$ defines the quantized disparity range for D in the $N_{\text{row}} \times N_{\text{col}}$ disparity map, and Q is a quantization function. See Fig. 9 with $d_{\max} = 60$.

In [5], [44], a Hough transform is used to detect the road manifold in form of a straight line in the v-disparity map. A more efficient and noise-resistant approach is to locate the dominating line following a stochastic process known as *random sample consensus* (RANSAC) [45]. The process first selects two bins randomly from the histogram, and a line hypothesis is solved $(\hat{m}, \hat{b})$. As values in the map define a density distribution, fitness of the hypothesis can be determined by summing up all the entries in $V(y, d)$ that are considered in the line up to a tolerable deviation (i.e. those inlier). Such a process is repeated for a finite number of iterations and that hypothesis, which achieves the highest fitness, is considered to be the dominating line.

As the precision of a line hypothesis is limited by the grid resolution, one may optionally perform weighted line fitting based on all the inliers to further improve the estimation.

C. Curve Fitting

The previously presented line fitting method can only handle planar road surfaces [31]. For a non-flat road geometry, the v-disparity map shows a curved distribution of pixel disparities. In [5], such a curve is approximated by a piecewise linear function, which is denoted by the envelop of straight lines corresponding to the k-strongest peaks in the Hough space, with $k \geq 1$ a chosen parameter.

In more recent work [20], [46], the curve is modelled by a 3^{rd} order B-spline or 2^{nd} order polynomial function. Adopting the polynomial model, the ground manifold estimation problem is solved by finding the coefficients of a polynomial $f(y)$ of degree n that best fits the curve in the v-disparity map:

$$d = f(y) = a_n y^n + a_{n-1} y^{n-1} + \ldots + a_1 y + a_0 \qquad (11)$$

where $a_0, a_1, \ldots, a_n$ are the coefficients, and the degree $n > 1$ is selected according to accuracy requirements for the algorithm.

Similar to the line-fitting technique, the fitness of a curve is defined by summing up all the curve's containing entries in V [10].

In order to generate the coefficients of the polynomial according to the degree specified, we need to compute a least-square polynomial for a given set of data. Following the least-square principle, we obtain the parameters $a_0, a_1, \ldots, a_n$, which minimize the total square error:

$$E(a_0, a_1, \ldots, a_n) = \sum_{i=1}^{m} [y_i - P(x_i)]^2 \qquad (12)$$

where $m \geq n$ is the number of samples. The optimal coefficients can be solved linearly.

D. Dynamic Programming and Graph Cut

Curve models with higher degrees provide flexibility to model a road manifold in v-disparity space. The degree of freedom is still limited by the adopted parametric model. Furthermore, curve models do not guarantee monotonicity that is often desired, as the depth of a road manifold does in general not increase as the row index goes from y to $y + 1$ (i.e. downward in the image). Following a discrete formulation, the curve fitting process is essentially a graph cut problem, which aims at finding a set of quantized disparities $\mathbf{d} = \{d_1, d_2, \ldots, d_{N_{\text{col}}}\}$ that minimizes a cost function subject to smoothness constraints.

Such a cut $\mathbf{d}$ divides the v-disparity map into left and right parts. To find the lower bound of the road manifold, the cost function can be defined by using a first-order derivative V_y of the v-disparity map V (i.e. along row y) [47]:

$$E(\mathbf{d}) = \sum_{y=1}^{N_{\text{col}}} V_y(y, d_y) + p \sum_{y=2}^{N_{\text{col}}} \Theta(d_{y-1}, d_y) \qquad (13)$$

where $p \geq 0$ defines a penalty for Θ, the smoothness function.

The value of p depends on the scale of the data term. To ensure the monotonicity of a cut, the smoothness term can be specified by an asymmetric L_1 Potts model:

$$\Theta(d_i, d_j) = \begin{cases} \infty, & \text{if } d_i > d_j \\ d_j - d_i, & \text{otherwise} \end{cases} \qquad (14)$$

Based on dynamic programming, an optimal cut can be solved using the Viterbi algorithm [48].

V. Multiocular Vision

The idea of the v-disparity space can be generalised to a multiocular camera set-up. As disparity spaces, derived from different stereo pairs, are not consistent to each other, the disparities have to be converted first into a universal representation (e.g. by using inverse-depth). Alternatively, one of the disparity spaces may be chosen as a reference such that all the disparities can be transformed and integrated appropriately.

In [20] a trinocular implementation is proposed for a generalization of the v-disparity map for three binocular stereo pairs defined by three cameras; Fig. 10 shows a trinocular data example from the KITTI *road* dataset [49]. Our extension is based on *transitivity error analysis in disparity space* (TED) as introduced in [50]. The approach is briefed as follows.

A disparity map $D : \Omega \to [0, d_{\max}]$ maps each pixel $(x, y) \in \Omega$ from the left image domain Ω to $(x - D(x, y), y)$ into the right image. A disparity map defines therefore a warping function $\mathcal{M} : \Omega \to \mathbb{R}$ as follows:

$$\phi(\mathcal{M}, D)(x, y) = \mathcal{M}\left(x - D(x, y), y\right) \qquad (15)$$

Given a collinear m-camera configuration, there are $m(m-1)/2$ left-right stereo pairs. The warping function ϕ can be used to construct the concatenation of any two disparity maps, following

$$\tau(D_{ij}, D_{jk})(x, y) = D_{ij}(x, y) + \phi(D_{jk}, D_{ij})(x, y) \qquad (16)$$

where $1 \leq i, j, k \leq m$. This concatenation defines the *TED-based disparities*.

A *TED-based error measure* can now be defined as

$$d_{ik,ijk}(x, y) = \|\tau(D_{ij}, D_{jk})(x, y) - D_{ik}(x, y)\| \qquad (17)$$

with respect to camera sequence (i, j, k). Function $d_{ik,ijk}$ measures the difference between an explicitly computed disparity map D_{ik} and the concatenated one $\tau(D_{ij}, D_{jk})$.

To apply TED to build a v-disparity map with respect to a camera pair, say $(0, 2)$, a *trinocular confidence measure* is defined:

$$\Gamma(x, y) = \frac{1}{1 + \|\tau(D_{01}, D_{12}) - D_{02}(x, y)\|} \qquad (18)$$

and a TED-weighted v-disparity map is constructed following

$$V(y, d) = \sum_{1 \leq x \leq N_{\text{col}} \ \wedge \ Q(D_{01}(x,y)) = d} \Gamma(x, y) \qquad (19)$$

Here, elements with higher TED-based confidence become more influential in the weighted v-disparity map, which can then be processed using again the described line fitting, curve fitting, or dynamic programming techniques.

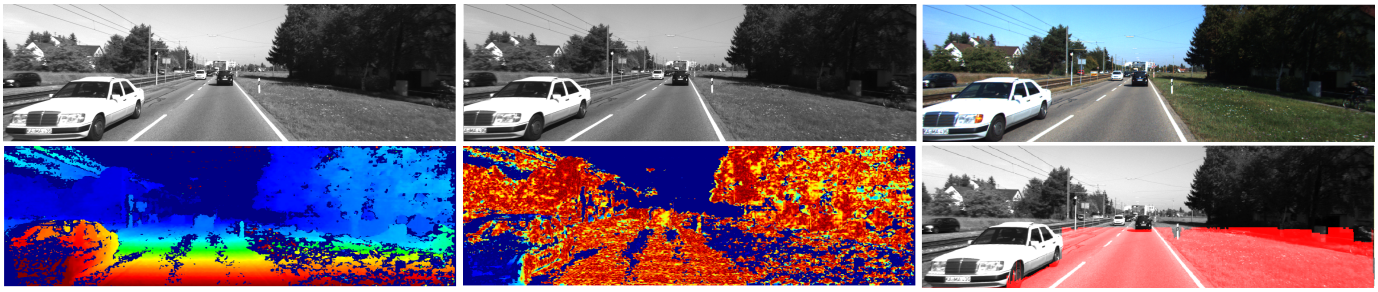

Fig. 10. Trinocular confidence and free space. *Top row*: Trinocular stereo pair from the KITTI *road* dataset. *Bottom left*: TED-based disparity. *Bottom middle*: Red and blue pixels indicate high and low confidence values, respectively. *Bottom right*: Calculated free-space (using v-disparity, confidence map, and proposed trinocular graph-cut).

VI. Experiments

We implemented the stixel construction process for four different disparity-based ground-manifold models as introduced above.[3] The base-line stixel method is implemented by mapping disparities into occupancy grids. Such a scheme suffers from shortages highlighted in [6]. Accordingly we selected four recently discussed models (plane-fit and line-fit [43], poly-fit [42], and graph-cut [47]) which are mainly dependent on the v-disparity space. This brings numerous advantages for ground-manifold detection. It is stated in [6] that working with original image coordinates, identified by the v-disparity space, is more practical when including probabilistic densities into the used model. Using the v-disparity space suppresses additional quantization artifacts, which is an arising problem when mapping measurements in Euclidean space into a grid or voxel space. Line or curve models are (still) dominant when using the v-disparity space for ground-surface estimation [53], thus also (still) dominating current stixel calculations [4], [23], [24].

Following [47], the number of missing stixels is used as an indicator for showing robustness when using the graph-cut approach. In this paper we extended the idea of using the graph-cut approach by including one more camera (i.e. a trinocular setup) utilizing the confidence map derived from TED. Furthermore, the experimental evaluation reported in this paper is more comprehensive than in [47] by also using LiDAR data and a number of statistical measures (more details later).

The computation of our disparity maps is based on the *Computer Vision System Toolbox* by calling a wrapped semi-global block matcher from the `OpenCV 3.1.0` library. In this section we report about the evaluation of detected stixels when applying one of those listed four ground-manifold models, and also when deciding either for binocular or trinocular recording, tested on $3,861$ frames. The evaluation is done using two widely-adopted datasets in the field, namely Daimler's 6D Vision Dataset,[4] and the KITTI Vision Benchmark Suite.[5]

A. Different Ground Manifold Models on 6D Vision Dataset

We evaluate the performance of stixel extraction for the following four ground manifold models: plane-fitting, line-fitting, polynomial-fitting, and graph-cut. The extracted stixels are verified on binocular stereo-image sequences downloaded from Daimler's 6D Vision website [54].

We applied the verification to all the twelve sequences which consist of $2,988$ 10-bit gray-scale stereo frames. The first six sequences are from the GOOD_WEATHER category, which present fairly good driving conditions with different illuminations, a variety of road views, shades, and colourings. The other six sequences from the BAD_WEATHER category present more challenging conditions such as rain drops, operating wind-shield wipers, and limited visibility.

In our work we compare extracted stixels with labelled frames provided by the dataset, and calculate a number of statistical measures. The *positive predictive value* (*PPV*), also known as *precision*, is calculated as

$$PPV = \frac{TP}{TP + FP} \tag{20}$$

where *TP* and *FP* denote the numbers of true positives and false positives, respectively. The *true positive rate* (*TPR*), also known as the *recall rate*, is defined as

$$TPR = \frac{TP}{P} \tag{21}$$

where $P = TP + FN$ is the number of positive pixels in the ground truth. We also calculated the *accuracy* (*ACC*) following

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{22}$$

where *TN* and *FN* denote the numbers of true negative and false negative pixels.

For those true positive pixels, we further evaluate the deviation of the disparities of the corresponding stixels against the ground truth. The root-mean-squares of the errors (RMSE) are also listed. These results are tabulated in Table I, with the best true positive rate in each sequence marked in bold.

It is found that all the models show low positive predictive values, ranging from 0.12 to 0.53. Further investigation reveals that the reason is due to high false-positive responses. In many cases, a detected stixel is not annotated in the test sequence. Although stixel ground truth was provided, they were annotated using a corridor[6] instead of the free-space, as it was observed during our experiments. An example is shown in Fig. 11.

[3]Implementation is in MATLAB R2017A.
[4]See www.6d-vision.com.
[5]See www.cvlibs.net/datasets/kitti/.
[6]The corridor is a subset of the free-space, and it denotes the region where the ego-vehicle is expected to drive in [8].

TABLE I
EVALUATION OF STIXEL EXTRACTION USING VARIOUS GROUND MANIFOLD MODELLING ON THE DAIMLER 6D-VISION DATASET

Sequence	Plane-fit				Line-fit				Poly-fit				Graph-cut			
	PPV	TPR	ACC	RMSE	PPV	TPR	ACC	RMSE	PPV	TPR	ACC	RMSE	PPV	TPR	ACC	RMSE
Seq. 1	0.44	**0.69**	0.92	1.66	0.42	0.63	0.92	1.52	0.41	0.63	0.92	1.51	0.40	0.64	0.92	**1.49**
Seq. 2	0.12	0.62	0.82	2.33	0.14	0.74	0.82	2.03	0.13	**0.74**	0.81	**1.99**	0.14	0.74	0.82	2.05
Seq. 3	0.47	0.74	0.82	**2.45**	0.50	0.74	0.83	2.46	0.49	0.68	0.83	2.69	0.50	**0.74**	0.83	2.53
Seq. 4	0.47	0.89	0.89	3.06	0.51	0.91	0.91	3.03	0.53	0.91	0.91	**3.02**	0.52	**0.91**	0.91	3.05
Seq. 5	0.22	0.94	0.80	2.20	0.23	**0.95**	0.80	**2.15**	0.23	0.89	0.81	2.40	0.23	0.92	0.81	2.18
Seq. 6	0.34	0.94	0.84	1.99	0.37	**0.95**	0.86	1.85	0.37	0.90	0.86	1.91	0.37	0.95	0.86	**1.85**
Average	0.34	0.80	0.85	2.28	0.36	0.82	0.86	**2.18**	0.36	0.79	0.86	2.25	0.36	**0.82**	0.86	2.19
Seq. 7	0.28	**0.47**	0.89	3.36	0.28	0.43	0.90	**3.36**	0.27	0.43	0.89	3.44	0.29	0.46	0.90	3.41
Seq. 8	0.23	0.80	0.87	4.12	0.24	0.81	0.88	3.93	0.25	0.82	0.89	4.02	0.26	**0.83**	0.89	**3.92**
Seq. 9	0.23	0.41	0.88	3.86	0.23	0.26	0.90	3.70	0.22	0.32	0.90	3.66	0.26	**0.44**	0.89	**3.58**
Seq. 10	0.26	0.76	0.81	2.90	0.25	0.65	0.82	2.91	0.28	0.78	0.82	2.82	0.28	**0.84**	0.82	**2.82**
Seq. 11	0.28	0.76	0.83	4.62	0.31	0.74	0.85	4.22	0.31	0.78	0.85	**4.19**	0.32	**0.81**	0.85	4.22
Seq. 12	0.27	0.58	0.91	3.62	0.25	0.34	0.92	3.54	0.28	0.50	0.92	3.53	0.29	**0.63**	0.91	**3.29**
Average	0.26	0.63	0.87	3.75	0.26	0.54	0.88	3.60	0.27	0.60	0.88	3.61	0.28	**0.67**	0.88	**3.54**

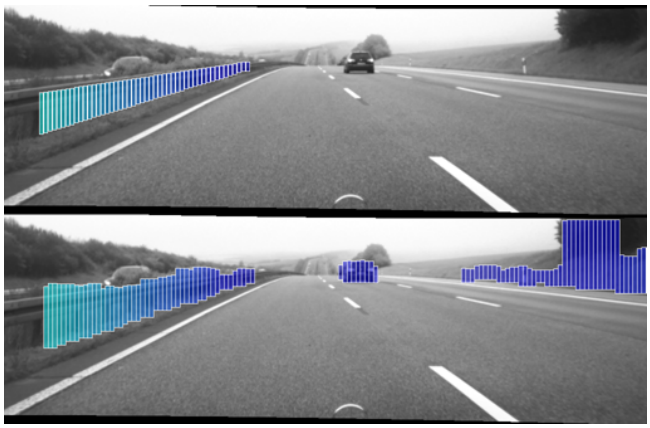

Fig. 11. Annotated ground truth (top) and extracted stixels (bottom) of the first frame of Sequence 1 from the 6D Vision dataset. The ramp on the right and the car are not annotated by the ground truth but detected by stixel implementation (poly-fit).

We therefore use the *recall rate* (*TPR*) as the major index to evaluate the ground-manifold models.

The four tested models perform similar for the GOOD_WEATHER category. The best recall rate average is achieved for the graph-cut model, which is just 2% better than the worst case - the plane-fit model. An overall accuracy around 0.86 is consistently found among all models, and the *RMSE* in disparities is between 2.18 to 2.28 pixels.

In the BAD_WEATHER category, however, distinctive results are found. In five out of six tested sequences, the graph-cut model achieves the best recall rate, which is 30% better than the worst rates in some extreme cases (Sequences 10 and 12); the graph-cut model is here followed by the poly-fit, plane-fit, and line-fit models. In general it is observed that the ground manifold cannot be effectively modelled by the line-fit method due to severely corrupted disparity maps under bad weather conditions.

An overall accuracy of about 0.88 is consistently found among all the models, and the *RMSE* in disparities is between 3.60 to 3.54 pixels.

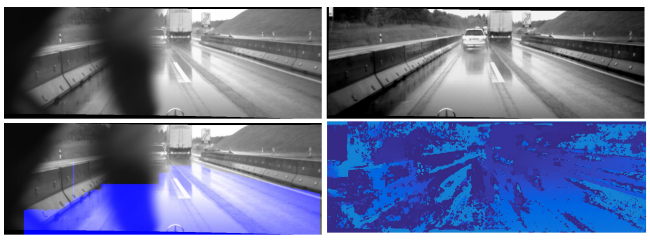

Fig. 12. Left image with window wiper(top-left) and right image (top-right) - frame number 142 of Sequence 11 (bad weather) from the 6D Vision dataset. The ground-manifold detection using binocular graph-cut (bottom-left). (Bottom-right) is showing disparity map for this challenging scene.

TABLE II
RUN-TIME PROFILING FOR STIXEL EXTRACTION USING VARIOUS GROUND-MANIFOLD MODELS ON THE DAIMLER 6D-VISION DATASET

Category	Plane-fit	Line-fit	Poly-fit	Graph-cut
GOOD_WEATHER	0.356 s	0.327 s	**0.326** s	0.332 s
BAD_WEATHER	0.452 s	**0.411** s	0.418 s	0.418 s

We also profiled the run-time for each model and show the average processing time per frame in Table II. The line-fit, poly-fit, and graph-cut models show similar computational time costs with a difference of not more than 5 milliseconds. The poly-fit yields the fastest approach for GOOD_WEATHER because it is insensitive to slope changes which widely exist in Sequence 1 (see Fig. 11). The plane-fit model is found to be most time consuming due to the iterative RANSAC process over a large amount of 3D data.

B. Comprehensive Evaluation on KITTI Dataset

We evaluate the quality of stixels not only for the selected four ground-manifold models, but also for binocular versus trinocular recording, using the trinocular data provided on the KITTI Vision Benchmark Suite [49].

Regarding previously stated challenges in evaluating stixels using the KITTI dataset [22], we address those by making use of the Velodyne high-definition 3D laser scanner data provided by the KITTI dataset. We use those range data as a ground-truth reference to evaluate the distance values assigned to the extracted stixels. This comprises of several processes:

1) Generate a disparity map from extracted stixels. The map contains valid disparities only for pixels belonging

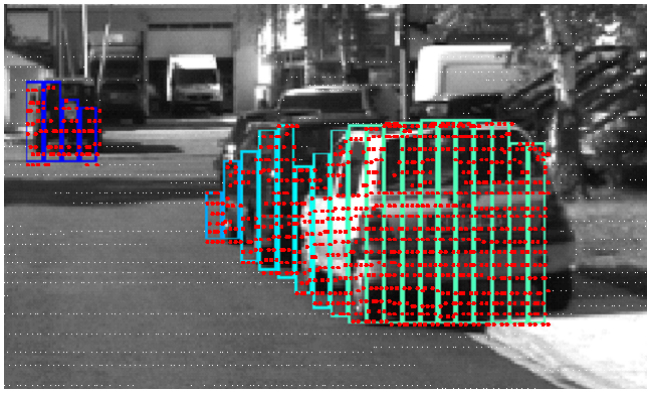

Fig. 13. Extracted stixels (color-coded by depth) and LiDAR points marked by white and red dots. Points hitting any extracted stixel are shown in red and used to evaluate the accuracy of the extraction process.

to a stixel. The map is then converted into a depth map following Eq. (5).

2) Project LiDAR points into image coordinates. Figure 13 shows some exemplary LiDAR point projections. The projections associate a subset of LiDAR range data to the extracted stixels.
3) For each associated LiDAR point, its depth is compared with the stixel depth map. The signed difference is then used to evaluate the performance of the stixel extraction process.
4) As the extracted stixels are in rectangular shape with reduced spatial resolution, it is often found close to edges of a stixel that background LiDAR points are wrongly assigned to a stixel. To exclude such outliers from the evaluation, we ensure a zero-mean for the error distribution of each model. Then, we discard LiDAR points that are outside the interval $[-0.5\sigma, +0.5\sigma]$ of all the range data associated to the same stixel before calculating the mode (note: not the mean) and the standard deviation.

We selected 873 trinocular stereo frames from the ROAD, RESIDENTIAL, and CITY categories, which include cars, cyclists, pedestrians, trees, and traffic signals. The test sequences are listed in Table III, also called for short A = `2011_09_26_drive_0032`, B = `2011_09_26_drive_0035`, and C = `2011_09_26_drive_0091` in the following tables.

TABLE III
SELECTED TEST SEQUENCES FROM THE KITTI DATASET

Category	Sequence	Frames
ROAD	`2011_09_26_drive_0032`	390
RESIDENTIAL	`2011_09_26_drive_0035`	137
CITY	`2011_09_26_drive_0091`	346

Qualitative results are listed in Table IV using a binocular configuration. We also use frames captured by the third camera to conduct additional tests on binocular versus trinocular stixels. Bold numbers indicate the best case per group, and colored numbers are the best case over all the seven models. Note that the plane-fit model is not of relevance here. As illustrated, a negative value means that laser points are in front of the stixels. Furthermore, as there are many non-flat objects present in the scene, and many background points are covered by the extracted stixels, we expect to see large standard deviation values.

For the ROAD sequence, the trinocular line-fit model achieves the lowest rate of a LiDAR-stixel error of -5.3cm, which is 55.5% better than the worst case yielded by the plane-fit model -11.6 cm. The main reason for this achievement is due to open-road scenarios which normally correspond closely to a straight-line in $v-$disparity space supplemented by the confidence measure using TED. This is slightly different compared to trinocular poly and graph-cut which achieve -6.5 and -6.2cm respectively.

In the RESIDENTIAL sequence, the used data show cars parked on the side of the road, houses, and road junctions. Based on the experiments, more obstacles (impacting the v-disparity map) make identifying a curve (using line-fitting or poly-fitting) more complicated. For this sequence, the trinocular graph-cut model has superior performance with a lowest mean LiDAR-stixel error of -10.2 cm. The disparity map relatively suffers from low-depth in this dataset due to lighting conditions accompanied with many pedestrians and buildings in the scenes. The performance of graph-cut is better suited for cases where there are irregular changes in a piecewise linear curve.

On the other hand, the binocular poly-fitting model provides the lowest mean LiDAR-stixel error of -3.5 cm for the CITY sequence as there are a number of non-flat objects in this sequence. This defines only a slight difference compared to the other techniques.

In addition to the statistics for the LiDAR-stixel error, we also calculate the improvement by the use of the third camera applying TED-weighted v-disparities (see Section V) as input for ground-manifold modelling.

As illustrated in Table V, the trinocular graph-cut approach covers more valid disparities compared to others, and appears to be insensitive to weather changes. It outperforms the trinocular polynomial or line-fit methods regarding robustness. The improvement rate is obvious for the ROAD and RESIDENTIAL sequences when using the graph-cut model. We notice that using trinocular cameras, the performance of poly-fit and line-fit decreases for RESIDENTIAL. This occurs because disparity values fluctuate roughly at the end of the data sequence (Frame 100 and onwards) because of having a round-about in the shown scenes. There are some values missing between $D01$ and $D12$ and this is reflected in values $\Gamma(x, y)$ since they are derived from these maps. The graph-cut model pays more attention to the disparity values, and using a penalization scheme is thus still able to recover the most relevant values compared to the ground manifold. The graph-cut model yields the highest improvement for ROAD and RESIDENTIAL with the trinocular configuration, and it still has promising results. This shows that, with such an extension, we can have a robust ground-manifold detection, resulting in accurate stixel estimation.

Finally, we summarise in Table VI the average number of stixels extracted per frame using binocular and trinocular vision-based ground manifold models. As shown, the binocular plane-fit performs best on the RESIDENTIAL sequence with an

TABLE IV
LIDAR-BASED QUALITATIVE EVALUATION [CM] OF GROUND MANIFOLD MODELLING USING KITTI DATASET (BINOCULAR AND TRINOCULAR CONFIGURATION).

Sequence	Binocular stereo								Trinocular stereo					
	Plane-fit		Line-fit		Poly-fit		Graph-cut		Line-fit		Poly-fit		Graph-cut	
	Mode	Std.dev.	Mode	Std. dev.	Mode	Std. dev.	Mode	Std. dev.	Mode	Std. dev.	Mode	Std. dev.	Mode	Std. dev.
A	-11.6	49.9	**-6.6**	54.9	-10.2	53.0	-9.5	54.2	-5.3	56.5	-6.5	**55.5**	-6.2	55.7
B	-14.4	54.1	-11.8	53.5	-12.5	54.1	**-10.9**	52.0	-12.9	53.8	13.0	53.7	-10.2	**52.9**
C	-5.1	47.4	-4.0	48.7	-3.5	50.2	-3.8	50.5	-4.2	**48.5**	**-3.8**	49.6	-3.9	50.1

TABLE V
IMPROVEMENT RATE WITH TRINOCULAR GROUND MANIFOLD MODELLING USING KITTI DATASET

Sequence	Line-fit			Poly-fit			Graph-cut		
	Mode	Std. dev.	Improve	Mode	Std. dev.	Improve	Mode	Std. dev.	Improve
A	-5.3	56.5	19.7%	-6.5	55.5	36.3%	-6.2	55.7	34.8%
B	-12.9	53.8	-10.2%	-13.0	53.7	-4.0%	-10.2	52.9	6.4%
C	-4.2	48.5	-5.0%	-3.8	49.6	-8.6%	-3.9	50.1	-2.6%

TABLE VI
AVERAGE NUMBER OF STIXELS EXTRACTED PER FRAME IN THE TESTED KITTI SEQUENCES

Sequence	Binocular stereo				Trinocular stereo		
	Plane-fit	Line-fit	Poly-fit	Graph-cut	Line-fit	Poly-fit	Graph-cut
A	32.6	32.2	33.8	34.2	35.0	**35.3**	34.8
B	**69.1**	27.3	24.3	29.1	29.7	26.7	28.7
C	66.9	71.0	69.5	70.7	**71.7**	71.0	70.6

average of 69.1% stixels detected. On the ROAD sequence, the trinocular polynomial-fit method yields the best result with an average of 35.3% stixels detected. The line-fit model achieved the best result on the CITY sequence with an average of 71.7% stixels detected per frame.

VII. CONCLUSION

This paper presented an in-depth analysis for binocular and trinocular vision-based stixel calculations using four ground-manifold models across two challenging datasets. For a comprehensive comparison, we provided an insight into the accuracy of extracted stixels on long-run sequences (for a total of $3,861$ frames); we also provided a brief run-time profiling to illustrate the performance of these models. The main objective of the reported research was to present an analysis on adopting a low-cost architecture (ground-manifold estimation method) for reducing false-positives in stixel estimations. Also, we extended the graph-cut model for a trinocular configuration which yields obvious and robust improvements compared to other models.

In our analysis we covered the number of cameras required and the road profile for obtaining accurate stixels. Experiments show for the binocular case, that the graph-cut model (using dynamic programming) presents a promising technique to ensure accuracy of stixels for the 6D vision and KITTI datasets. The number of true-positives is large when the graph-cut model is used as a minimisation method for calculating a v-disparity cut; see results for the 6D vision dataset for the GOOD_WEATHER as well as the BAD_WEATHER categories. As illustrated, the polynomial-fit model shows the fastest run-time for GOOD_WEATHER, while the line-fit model achieves the fastest run-time for BAD_WEATHER.

In order to evaluate the effects for the KITTI dataset, a comprehensive study was conducted not only for comparing ground-manifold models but also bi- versus trinocular recording. Results show that the number of generated stixels highly increases when using trinocular line fitting for ROAD sequences, and binocular poly-fitting for CITY sequences; finally, trinocular graph-cut proved to be the best alternative on RESIDENTIAL sequences. Having especially challenging scenes in mind, altogether we recommend the trinocular graph-cut approach.

REFERENCES

[1] Montani, C. and Scopigno, R.:Rendering volumetric data using STICKS representation scheme. In *ACM SIGGRAPH Computer Graphics*, 24(5): 87–93, 1990.

[2] Badino, H., Franke, U., and Pfeiffer, D.: The stixel world - A compact medium level representation of the 3D-world. In Proc. *DAGM*, LNCS 5748, 51–60, 2009.

[3] Klette, R.: Vision-based driver assistance. In: *Wiley Encyclopaedia Electrical Electronics Engineering*. John Wiley & Sons, 1–15, 2015.

[4] Seo, J., Oh, C., and Sohn, K.: Segment-based free space estimation using plane normal vector in disparity space. In Proc. *Connected Vehicles Expo*, 144–149, 2015.

[5] Labayrade, R., Aubert, D., and Tarel, J.: Real time obstacle detection in stereovision on non flat road geometry through v-disparity representation. In Proc. IEEE Symp. *Intelligent Vehicles*, 646–651,2002.

[6] Pfeiffer, D.: The Stixel World. Doctoral Thesis, Humboldt Universität Berlin, 2011.

[7] Badino, H., U. Franke, and R. Mester: Free space computation using stochastic occupancy grids and dynamic programming. In Proc. *ICCV Workshop Dynamical Vision*, 2007.

[8] Shin, B.-S., Xu, Z., and Klette, R.: Visual lane analysis and higher-order tasks: A concise review. *Machine Vision Applications*, 25(6): 1519–1547, 2014.

[9] Onoguchi, K., N. Takeda, and M. Watanabe: Obstacle location estimation using planar projection stereopsis method. *Systems Computers Japan*, 32(14): 67–76, 2001.

[10] Saleem, N. H. and Klette, R.: Accuracy of free-space detection for stereo versus monocular vision. In Proc. *Image Vision Computing New Zealand*, 48–53, 2016.

[11] Rezaei, M., and Klette, R.: *Computer Vision for Driver Assistance - Simultaneous Traffic and Driver Monitoring*. Springer, Amsterdam, Series "Computational Imaging and Vision", Volume 45, 2017.

[12] Kaaniche, K., Demonceaux, C., and Vasseur, P.: Analysis of low-altitude aerial sequences for road traffic diagnosis using graph partitioning and Markov hierarchical models. In Proc. *Int. Multi-Conf. Systems Signals Devices*, 656–661, 2016.

[13] Wu, J., Cui, Z., Sheng, V. S., Zhao, P., Su, D., and Gong, S.: A comparative study of SIFT and its variants. *Measurement Science Review*, 13 (3): 122–131, 2013.

[14] Farabet, C., Couprie, C., Najman, L., and Lecun, Y.: Learning hierarchical features for scene labeling. *IEEE Trans. Pattern Analysis Machine Intelligence*, 35 (8): 1915–1929, 2013.

[15] Anders, J., Mefenza, M., Bobda, C., Yonga, F., Aklah, Z., and Gunn, K.: A hardware/software prototyping system for driving assistance investigations. *J. Real-Time Image Processing*, 11 (3): 559–569, 2016.

[16] Hirschmüller, H.: Stereo processing by semiglobal matching and mutual information. *IEEE Trans. Pattern Analysis Machine Intelligence*, 30: 328–341, 2008.

[17] Guan, S., and Klette, R.: Belief-propagation on edge images for stereo analysis of image sequences. In Proc. *Int. Workshop Robot Vision*, LNCS 493, 291–302, 2008.

[18] Morales, S., Vaudrey, T., and Klette, R.: Robustness evaluation of stereo algorithms on long stereo sequences. In Proc. IEEE Conf. *Intelligent Vehicles*, 347–352, 2009.

[19] Klette, R.: *Concise Computer Vision*. Springer, London, 2014.

[20] Saleem, N., Chien, H.-J.,Rezaei, M., and Klette, R.: Improved stixel estimation based on transitivity analysis in disparity space. In Proc. *Computer Analysis Images Patterns*, LNCS 10424, 28–40, 2017.

[21] The Northland Transport Technology Testbed. www.n3t.kiwi , 2016.

[22] Sanberg, W. P., Dubbelman, G., and deWith, P. H. N.: Color-based free-space segmentation using online disparity-supervised learning. In Proc. IEEE Conf. *Intelligent Transportation Systems*, 906–912, 2015.

[23] Lee, S., Suhur, J. K., Jung, H. G.: Improvement of Stixel Segmentation Using Additive Image Domain Features and Genetic Algorithm-based Optimization. *Trans. Korean Society Automotive Engineers*, 565–574, 2015.

[24] Hernandez, D., Espinosa, A., Moure, J., Vázquez, D., López, A.: GPU-accelerated real-time stixel computation. In Proc. *Conf. Applications Computer Vision*, 1054–1062, 2017.

[25] Schneider, L., Cordts, M., Rehfeld, T., Pfeiffer, D., Enzweiler, M., Franke, U., Pollefeys, M., and Roth, S.: Semantic stixels: Depth is not enough. In Proc. IEEE Symp. *Intelligent Vehicles*, 110–117, 2016.

[26] Lu,K.,Li, J.,An X., and He, H.: A hierarchical approach for road detection. In Proc. *IEEE Int. Conf. Robotics Automation*, 517–522, 2014.

[27] He, Z.,Wu, T.,Xiao, Z., and He, H.: Robust road detection from a single image using road shape prior. In Proc. IEEE *Int. Conf. Image Processing*, 2757–2761, 2013.

[28] He, Y.,Wang, H., and Zhang, B. : Color-based road detection in urban traffic scenes. *IEEE Trans. Intelligent Transportation Systems*, 5 (4): 309–318, 2004.

[29] Kong, H., Audibert, J., and Ponce, J.: General road detection from a single image. *IEEE Trans. Image Processing*, 19 (8) : 2211–2220, 2010.

[30] Miksik, O.: Rapid vanishing point estimation for general road detection. In Proc. *IEEE Int. Conf. Robotics Automation*, 4844–4849, 2012.

[31] Suhur, J. and Jung, H.: Dense stereo-based robust vertical road profile estimation using Hough transform and dynamic programming. *IEEE Trans. Intelligent Transportation Systems*, 1528–1536, 2015.

[32] Perrollaz, M., Yoder, J.D., Negre, A., Spalanzani, A., and Laugier, C.: A visibility-based approach for occupancy grid computation in disparity space. *IEEE Trans. Intelligent Transportation Systems*, 13:1383–1393, 2012.

[33] Neumann, L., Vanholme, B. ,Gressmann, M.,Bachmann A.,Kahlke L., and Schule, F.: Free space detection: A corner stone of automated driving. In Proc. *IEEE Conf. Intelligent Transportation Systems*, 1280–1285, 2015.

[34] Harms, H., Rehder, E., and Lauer, M.: Grid map based free-space estimation using stereovision. In Proc. IEEE IV Workshop *Environment Perception Automated On-road Vehicles*, 2015.

[35] Benenson, R., Mathias, M., Timofte, R., and Van Gool, L.: Fast stixels estimation for fast pedestrian detection. In Proc. *ECCV*, LNCS 7585, 11–20, 2012.

[36] Levi, D., Garnett, N., and Fetaya, E.: StixelNet: A deep convolutional network for obstacle detection and road segmentation. In Proc. *British Machine Vision Conf.*, 1:12, 2015.

[37] Benenson, R., Timofte, R., and Van Gool, L.: Stixels estimation without depth map computation. In Proc. *ICCV Workshops*, 2010–2017, 2011.

[38] Scharwächter, T., and Franke, U.: Low-level fusion of color, texture and depth for robust road scene understanding. In Proc. IEEE Symp. *Intelligent Vehicles*, 599–604, 2015.

[39] Cordts, M., Schneider, L., Enzweiler, M., Franke, U., and Roth, S.: Object-level priors for stixel generation. In Proc. *German Conf. Pattern Recognition*, 172–183, 2014.

[40] Thrun,S., and Bü, A.: Integrating grid-based and topological maps for mobile robot navigation. In Proc. *Nat. Conf. Artificial Intelligence*, vol. 2, 944–950, 1996.

[41] Pfeiffer, D. and Franke, U.: Efficient representation of traffic scenes by means of dynamic stixels. In Proc. IEEE Symp. *Intelligent Vehicles*, 217–224, 2010.

[42] Saleem, N., Rezaei, M., and Klette, R.: Extending the stixel world using polynomial ground manifold approximation. In Proc. *IEEE Conf. Mechatronics Machine Vision Practice*, 526–531, 2017.

[43] Dhiman, A., Chien, H.-J., and Klette, R.: Road surface distress detection in disparity space. In Proc. *Int. Conf. Image Vision Computing New Zealand (IVCNZ)*, DOI: 10.1109/IVCNZ.2017.8402459, 2017.

[44] Iloie, A., Giosan, I., and Nedevschi, S.: UV disparity based obstacle detection and pedestrian classification in urban traffic scenarios. In Proc. *Intelligent Computer Communication Processing*, 119–125, 2014.

[45] Fischler, M., and Bolles, R.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. ACM*, 24 (6): 381–395, 1981.

[46] Schauwecker, K., Morales, S., Hermann,S., and Klette, R.: A comparative study of stereo-matching algorithms for road-modeling in the presence of windscreen wipers. In Proc. IEEE Symp. *Intelligent Vehicles*, 7–12, 2011.

[47] Saleem, N., Chien, H.-J., and Klette, R.: Stixel optimization: Representing challenging on-road scenes. In Proc. *Int. Conf. Image Vision Computing New Zealand (IVCNZ)*, DOI: 10.1109/IVCNZ.2017.8402446, 2017.

[48] Viterbi, A. J.: Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Information Theory*, 13(2): 260–269, 1967.

[49] Geiger, A., Lenz, P., and Urtasun, R.: Are we ready for autonomous driving? The KITTI vision benchmark suite. In Proc. *Computer Vision Pattern Recognition*, 3354–3361, 2012.

[50] Chien, H.-J. ,Geng, H. , and Klette, R.: Improved visual odometry based on transitivity error in disparity space: A third-eye approach. In Proc. *Image Vision Computing New Zealand*, 72–77, 2014.

[51] Davies, E.: *Machine Vision: Theory, Algorithms, Practicalities*, Morgan Kaufmann, 2004.

[52] Otsu, N.: A threshold selection method from gray-level histograms. *IEEE Trans. Systems Man Cybernetics*,9:62–66, 1979.

[53] Sivaraman, S. and Trivedi, M.: Looking at vehicles on the road: A Survey of vision-based vehicle detection, tracking, and behavior analysis. *IEEE Trans. Intelligent Transportation Systems*, 1773–1795, 2013.

[54] Pfeiffer, D., Gehrig, S., and Schneider, N.: Exploiting the power of stereo confidences. In Proc. *Computer Vision Pattern Recognition*, 297–304, 2013.

Noor Haitham Saleem worked as a lecturer at the Computer Science Institute, Slemani Polytechnic University, Kurdistan Region, Iraq, before pursuing his PhD project since 2015 at Auckland University of Technology, Auckland, New Zealand. His research interests include vision-based driver assistance systems and multimedia application developments. In 2016 he had a visiting position at Wuhan University, China, for testing trinocular stereo in vehicles. He received the "Best Student Paper Award" at CAIP 2017 in Sweden. At ICCAR 2018 in Auckland, he received the "Best Presentation Award". His accepted paper at IVCNZ 2017, Christchurch, had been "upgraded" to a keynote at this conference.

Hsiang-Jen Chien received his PhD degree from the Department of Electrical and Electronic Engineering at Auckland University of Technology, New Zealand, in 2017. He has been working in the field of computer vision since 2007, especially on shape recovery, 3D reconstructions, range data fusion, and visual odometry. He worked as a full-time research developer on clinical imaging technologies in the medical industries of Taiwan, and also on one of the pioneering LiDAR-vision mobile sensing systems in Taiwan before he commenced his PhD study in Auckland, New Zealand, in 2014. During his PhD project he had a six-months internship in the "Environment Perception - Image Understanding Group", Research and Development, Daimler A.G., Germany, under the supervision of Dr. Uwe Franke. Dr. Chien received various honours and awards at international conferences.

Mahdi Rezaei is a Senior Lecturer at the Computer Science Department, Qazvin Islamic Azad University, Iran, and also an Honorary Researcher at Auckland University of Technology, Auckland, New Zealand. He received his PhD degree in 2014 at the University of Auckland with a top award for PhDs in the Computer Science Department of this university in 2014. He is the leading author of the monograph "Computer Vision for Driver Assistance - Simultaneous Traffic and Driver Monitoring", published by Springer. Amsterdam, in 2017. He has published at leading conferences such as CVPR, PSIVT, ACCV, and CAIP. His research interests are in general in computer vision, artificial intelligence, and robotics, and in particular in machine learning, object detection, classification, and autonomous vehicles. He is the head of a national project (of the Iran Ministry of Transportation) on automatic number plate recognition.

Reinhard Klette is a Fellow of the Royal Society of New Zealand and a professor at Auckland University of Technology, Auckland, New Zealand. He was the founding Editor-in-Chief of the Journal of Control Engineering and Technology in 2011 to 2013 and an associate editor of IEEE Trans. PAMI in 2001–2008. Currently he is on the editorial boards of the International Journal of Computer Vision (as on honorary member) and the International Journal of Fuzzy Logic and Intelligent Systems. He (co-)authored more than 300 publications in peer-reviewed journals or conferences, and also books on computer vision, image processing, geometric algorithms, and panoramic imaging. He presents keynotes at international conferences since the 1990s. Springer London published in January 2014 his book entitled "Concise Computer Vision". In 2017 he received the Quancheng Friendship Award of Shandong Province, China, and became a Helmholtz International Fellow in Germany.