Monocular Stixels: A LIDAR-guided Approach

Noor Haitham Saleem, Anthony Griffin, and Reinhard Klette
School of Engineering, Computer and Mathematical Sciences
Department of Electrical and Electronic Engineering
Auckland University of Technology
Auckland, New Zealand

***Abstract*—Stixel calculations are commonly based on binocular vision; these calculations map millions of pixel disparities into a few hundred stixels. Depending on applied stereo vision, this binocular approach is sometimes incapable of dealing with low-textured road information or noisy data. The main objective of this work is to propose a more reliable approach to calculating stixels by incorporating laser scanners (i.e., LIDAR). We show that this supports more efficient and robust 3D point representations, even if only integrating monocular vision into the LIDAR-based approach for generating monocular stixels. Experimental results show a more accurate (by 15.4%) stixel detection rate when the LIDAR-guided monocular configuration is used compared to a conventional binocular approach.**

I. Introduction

Robust obstacle segmentation and scene understanding are key tasks for visual sensors (cameras) in autonomous cars in order to be able to interpret and act within a dynamic environment. Cameras play a significant role in autonomous driving; they are capable of providing rich information including distances to obstacles given in traffic scenes. Incorporating remote sensing (e.g., LIDAR) adds benefits to autonomous cars if it provides depth information at high accuracy. A high market growth is expected for LIDAR technologies for the next few years. Firms already advertise low-cost LIDAR sensors [1].

The use of cameras for stereo vision is an advanced field of research. The reduction of processing time has been investigated in relation to accurate representations of stereo data (e.g., by disparities) [2]. As a result, models of image content have emerged that represent raw stereo data while being neither too generic nor too specific. Stixels define such a model; a *stixel* (from "stick element") is a thin column in vertical pose of defined height on a base rectangle of fixed pixel width [3], [4].

Various approaches for stixel estimation have been investigated by mainly involving bi- or trinocular vision, since depth can be obtained from stereo cameras at low cost. A failure of disparity estimation on obstacle or low-textured road surfaces still causes concerns [5]. Unstable results caused by challenging imaging conditions (represented by illumination, colour, or texture) may be resolved by also using sensors (such as LIDAR) which are reliable under such conditions. As a result, this may lead to improved disparity maps. Yet, LIDAR points are sparse and there must be an optimized interpolation approach that would support us in our endeavour to obtain a dense depth map (see Fig. 1), and later a dense stixel representation. This research proposes monocular stixels guided by LIDAR data for verified stixel positions.

The rest of the paper is structured as follows: Section 2 introduces common methods for estimating stixels, Section 3 discusses the proposed methods, Section 4 provides our experimental analysis, and Section 5 concludes.

II. Related Work

Stixels are compact representations towards semantic segmentation. Neighbouring cells in an occupancy grid (e.g., above a $w \times w$ base) are at about the same depth; a stixel forms a vertical "stick" above its base [4]. Relevant studies on stixel detection can be categorized either as single-layer estimation or multi-layer segmentation.

Single-layer stixel extraction can have reduced computational costs; in [6], a direct stixel computation is presented by changing the parametrization from disparity space into a pixel-wise cost volume for speed improvement. In [7], the authors use monocular vision and deep convolutional neural networks for road detection, while stixel calculation is done using binocular vision. Computing stixels by using stereo images (i.e., depth cues) in combination with colour appearance was proposed to solve illuminance and texture problems that exist in binocular vision. Such methods have been presented for stixel segmentation [8]–[10]. Trinocular-based stixel estimation, proposed by [11], aimed at fitting a polynomial curve model to the ground manifold.

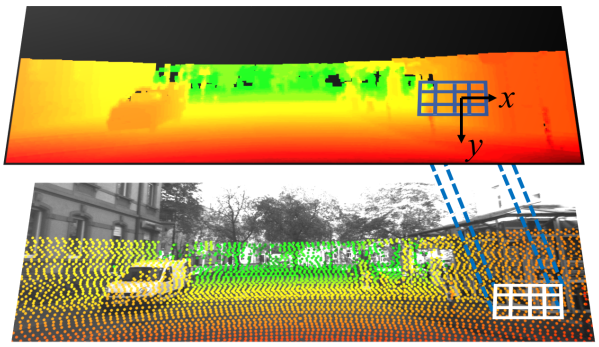

Fig. 1. *Top*: Dense disparity map (supporting a dense depth map). *Bottom*: Sparse 3D points measured by a LIDAR sensor.

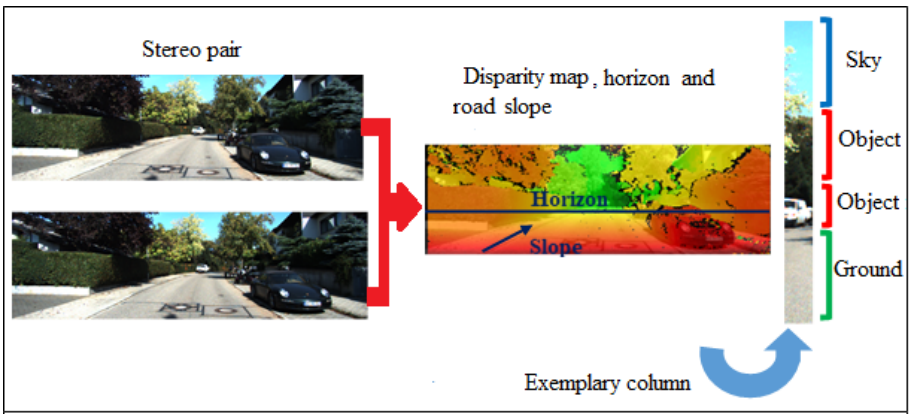

Fig. 2. Stixels describe obstacles: The figure shows the path from a stereo pair to one example of a column, to be mapped into four stixels, one on top of the one below.

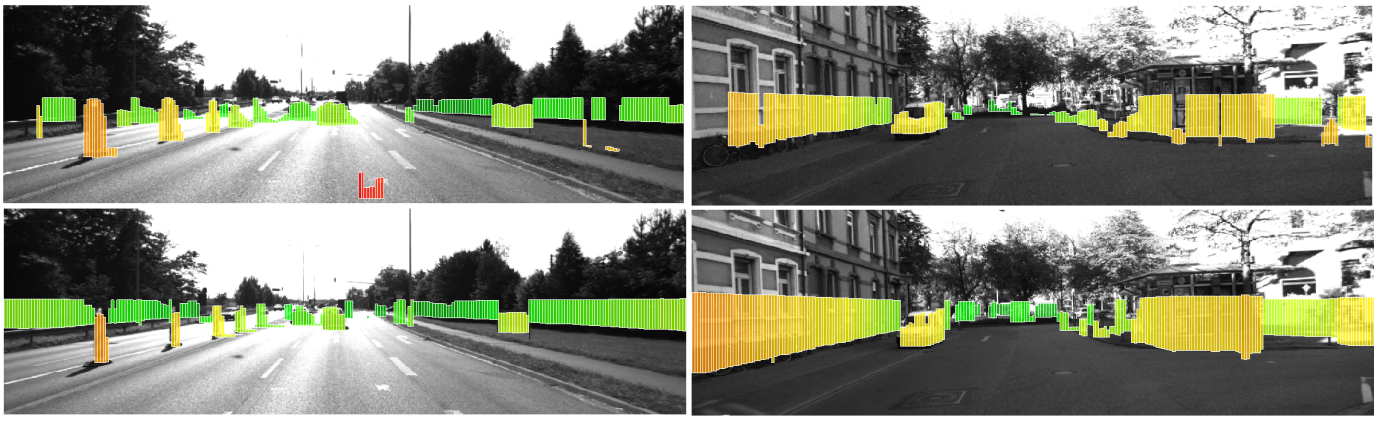

Fig. 3. Single-layer stixel estimation. *Top row*: Estimation using a disparity map resulting from stereo-matching (binocular). *Bottom row*: Estimation using a depth map incorporating a LIDAR sensor (monocular).

A single- or first-layer stixel representation of a scene (i.e., considering only at most one stixel per image column) has limitations for obstacle representations (e.g., it cannot represent a moving vehicle and a guardrail in the same column). This issue led [4] to extend the single-layer stixel model into a model with multi-layer stixels in each column (see Fig. 2) using a unified probabilistic approach. This approach uses dynamic programming applied to *v-disparities* to measure the occurrence of a certain class (i.e., object, sky, or ground) for multiple stixels per column.

The extended representation yields a highly efficient modelling of scene objects in urban traffic environments [9]. It is used as a complementary model for various autonomous driving applications such as object tracking, which demonstrate how the stixels' velocity is tracked over a time-stamp [4]. Furthermore, a GPU-based acceleration for stixel calculation is presented by [12]; the stereo-matcher used was based on GPU-acceleration of a dense stereo calculation using semi-global matching (SGM).

As observed during experiments, there are a lots of methods focused on designing a reliable stixel representation, however, since the input of that model suffers from noise then it can still degrade the accuracy of the detected stixels (see Fig. 3 which shows the difference between monocular and binocular stixels employing the same process). On the other hand, to recover multi-layer stixel segmentation, an adopted colour fusion model might not be suitable due to the shortcomings highlighted in [13]. Our approach mainly focuses on fusing LIDAR data to improve the accuracy of stixel calculation—for both single-layer and multi-layer—compared to conventional stereo based stixels.

III. Proposed Method

The proposed method aims to include LIDAR data in the estimation of monocular stixels. To do this, we first extract a disparity map from an distance map assuming a hypothetical second camera at an assumed base distance b, we then:

- Project 3D LIDAR points into the 2D image plane (point projection). The results improved by discarding points outside the camera plane (noisy points) and the remaining points are sorted according to their position in pixel units, in order to speed up the search process.
- Construct a dense distance map from sparse LIDAR points using color and texture information.

- Convert the distance map into a disparity map based on the camera matrix (we simply use focal length f and base distance b as reported for the used KITTI data.
- Construct stixels based on this "monocular) disparity" map following the common procedure as for binocular vision.

A. Point-projection Phase

The provided calibrated data (images and LIDAR points) in the KITTI dataset are the input used to obtain the dense map. This paper uses the spatial relationship between 3D points projected into the image plane to construct a dense map. As described by [14], the Velodyne HDL-64E S2 is employed in the KITTI dataset which has 0.09 degree angular resolution and 2 cm distance accuracy. It is efficient and able to collect around 1.3 million points/second. Scans are stored as floating points with $[x, y, z]$ coordinates in which x, y, z represent forward, to the left, and upward directions, respectively, using:

$$\mathbf{K}_{\text{velo}}^{\text{cam}} = [\mathbf{R}_{\text{velo}}^{\text{cam}} | \mathbf{t}_{\text{velo}}^{\text{cam}}] \tag{1}$$

where $\mathbf{R}_{\text{velo}}^{\text{cam}} \in \mathbb{R}^{3\times3}$ is the rotation matrix, and $\mathbf{t}_{\text{velo}}^{\text{cam}} \in \mathbb{R}^{3\times3}$ is the translation vector, in both cases of Velodyne sensor into camera pose.

Detailed information regarding LIDAR and camera calibration, data alignment, and the calibration matrices can be found in [14], and intrinsic and extrinsic parameters are given in [15]. A 3D point in LIDAR coordinates $P_r = [x, y, z, 1]^\top$ is projected into a point in camera coordinates $P_s = [x, y, z, 1]^\top$, based on:

$$P_s = \mathbf{K}_{\text{velo}}^{\text{cam}} P_r \tag{2}$$

Every point P_s is then rectified to match the image plane using a rectification matrix $\mathbf{K}_{\text{rec}}$:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{K}_{\text{rec}} P_s \tag{3}$$

Considering the projected LIDAR points in pixel coordinates (x, y), as given by (3), some operations are performed prior to the interpolation stage (described in the following section). The points outside the camera plane are discarded, and the remaining points are sorted according to their position in pixel units, in order to speed-up the search process. Finally, the points are rearranged into a new space that combines the coordinates in pixel units (x, y) and the range r, such that a point P is represented by $P = [x, y, r]^\top$.

B. Interpolation phase

The point clouds, provided by LIDAR, are sparse and in some cases noisy, and thus an interpolation method is required to derive a smooth (filtered) "dense" distance map. The interpolation process carried out is a combination of methods proposed by [16], [17] which both focused on color and texture information. Basically, these methods present a solution to sparse data by merging 3D points with information from RGB images. The assumption is based on the idea that pixels in a connected region, having similar texture in the camera image in their neighborhood, will have identical depth values.

Furthermore, [16] generates a [0-255] normalized depth map image from LIDAR data. This situation does not work for us since the stixel calculation requires a real depth, not a [0-255] normalized depth map. Points $P = [x, y, r]^\top$ represent a calibrated set of 3D sparse LIDAR points projected into the image plane, as described in the previous phase. In order to derive the distance (or range) map R at a given position (x, y), we can calculate this map by a weighted fusion of the range values r_k of the sparse points P in a window W_p centered at position $p = (x, y)$, as follows:

$$R(p) = \sum_{k \in W_p} \omega_k \cdot r_k \tag{4}$$

The window W_p is of size 5×5 in our experiments.

Even for the fixed-size window, the number k of points in W_p varies and depends on the 3D-cloud's sparsity. A similar mechanism is applied to a bilateral filter when interpolating low-resolution images; each weight ω_k is computed by two factors:

- a pixel distance function $d_2(p, q)$ (here: assumed to be the Euclidean distance in pixel units) between the window's central point $p = (x, y)$ and the considered k points $q = (i, j)$ within the window W_p, and
- a confidence weighting term $\kappa(r)$ which is determined as a function of the measured distance r. In some cases (e.g., uncertainty in sensor data), $\kappa(r)$ decreases linearly corresponding to the range value, penalizing 3D points in direct relation to their distance from the LIDAR. The $\kappa(r)$ values are normalized by the maximum range value r_k in W_p; see [16], [17].

Hence, the 2D spatial neighborhood filter is re-written as:

$$R(p) = \sum_{q \in W_p} d_2(p, q) \cdot \kappa(r_q) \cdot r_q \tag{5}$$

From the distance map, the calculation of the disparity map is as follows:

$$D(p) = f \cdot \frac{b}{R(p)} \tag{6}$$

where f is the focal length and b the (assumed) camera baseline; here we use $b = 0.54$ m as in the KITTI data.

C. Single-layer stixels

To construct a single layer stixel, we adopt the process outlined in [3], [18] but use the disparity map D that is derived from monocular vision. The disparity map D would be more accurate since we are fusing multiple sensors' information. As detailed in [18], an optimization approach was proposed to minimise the cost of a cut in v-disparity space to identify a piecewise linear curve. Following a discrete formulation, the curve fitting process is essentially a graph-cut problem, which aims at finding a set of quantised disparities $\mathbf{d} = \{d_1, d_2, \ldots, d_{N_{\text{col}}}\}$ that minimises a cost function subject to smoothness constraints. Such a cut $\mathbf{d}$ divides the v-disparity

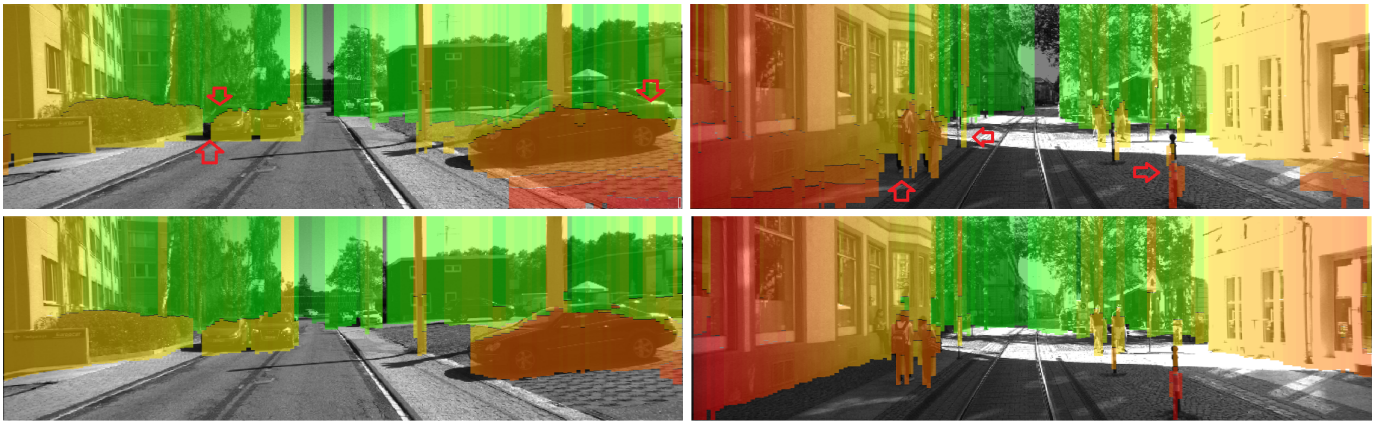

Fig. 4. Multi-layer stixel maps tested on KITTI data using depth obtained from LIDAR. *Top row*: Red arrows depict current multi-layer stixel problems (displacements in the representation). *Bottom row*: Examples when using the proposed solution.

map (row-wise) into left and right parts. To find the lower bound of the road manifold, the cost function (i.e., error or energy E) can be defined by using a first-order derivative V_y of the v-disparity map V (i.e., along row y) [18]:

$$E(\mathbf{d}) = \sum_{i=1}^{N_{\text{col}}} V_y(y, d_i) + \gamma \sum_{i=2}^{N_{\text{col}}} \Theta(d_{i-1}, d_i) \tag{7}$$

where $\gamma \geq 0$ defines a penalty for Θ, the smoothness function, and $V(y, d)$ represents the number of pixels sharing the same disparity of d in the y-th row of the disparity map D derived from LIDAR in (6). The value of γ 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 (more details in [18]).

D. Multi-layer stixels

An essential step in multi-layer stixel estimation is estimating the road surface. As presented in [4], the road surface can be estimated directly from camera parameters, however, this scheme might be infeasible when the provided dataset is missing some information about camera parameters (i.e., tilt angle). As shown in Fig. 4, the monocular stixel (first-row) is supposed to be improved but there are a still a lot of false positives. These affect obstacle representation and road surface estimation. Using a point cloud we need to estimate the road manifold from which we can derive a 3D rotation and translation matrix. Usually, converting a world coordinate $P_w = [X_w, Y_w, Z_w]^\top$ into an image plane coordinate requires geometric information such as:

$$\varepsilon \cdot [x, y, 1]^\top = \mathbf{K}[\mathbf{R} \,|\, \mathbf{t}][X_w, Y_w, Z_w]^\top \tag{8}$$

where x, y represent the image plane coordinates and ε is a depth scalar for depth values > 0. We can remove $[\mathbf{R} \,|\, \mathbf{t}]$ from the above equation since the rotation matrix is an identity matrix and the translation vector is all zeros. We can solve the above unknown parameters by applying matrix inversion:

$$\frac{1}{\varepsilon}[X_w, Y_w, Z_w]^\top = \mathbf{K^{-1}}[x, y, 1]^\top \tag{9}$$

We can represent the left side in (9) by a variable C. This equation can be then re-written as:

$$C = \mathbf{K^{-1}}[x, y, 1]^\top \tag{10}$$

To identify a pixel with world coordinates X_w, Y_w, Z_w, we need

$$[X_w, Y_w, Z_w]^\top = C \cdot \varepsilon \tag{11}$$

Then, we can use *M-estimator sample consensus* (MSAC) which fits a plane to a cloud of points. The fitting process is applied only on inlier points that have a maximum tolerable distance to the plane. The model can be verified in the road plane equation to estimate the road plane coefficients:

$$a_0 X_w + a_1 Y_w + a_2 Z_w + a_3 = 0 \tag{12}$$

So far we just estimate the road plane coefficients. In order to find a known world coordinate location in the depth map image, we use

$$C = \frac{1}{\lambda}[X_w, Y_w, Z_w]^\top \tag{13}$$

where λ is the ground depth scalar to be calculated. That means

$$[X_w, Y_w, Z_w]^\top = [\lambda C_1, \lambda C_2, \lambda C_3]^\top \tag{14}$$

By substituting Eq. (14) in Eq. (12), this results in:

$$a_0 \lambda C_1 + a_1 \lambda C_2 + a_2 \lambda C_3 + a_3 = 0 \tag{15}$$

The value of λ can be found by

$$\lambda = \frac{-a_3}{a_0 C_1 + a_1 C_2 + a_2 C_3} \tag{16}$$

Finally, the ground disparity can be calculated:

$$\text{GD} = f \cdot \frac{b}{\lambda C_3} \tag{17}$$

For multi-layer stixel construction, consider a disparity map D of size $N_{\text{col}} \times N_{\text{row}}$ in which each column x defines segments L_x describing classes in $\mathbb{C} = \{g, o\}$. Let $N_x \leq N_{\text{row}}$ be the total number of segments in column x. Formally,

$$\begin{aligned} L &= \{L_x : 1 \leq x \leq N_{\text{col}}\} \in \mathbb{L}, \text{ with} \\ L_x &= \{s_x^n : 1 \leq n \leq N_x\} \end{aligned} \tag{18}$$

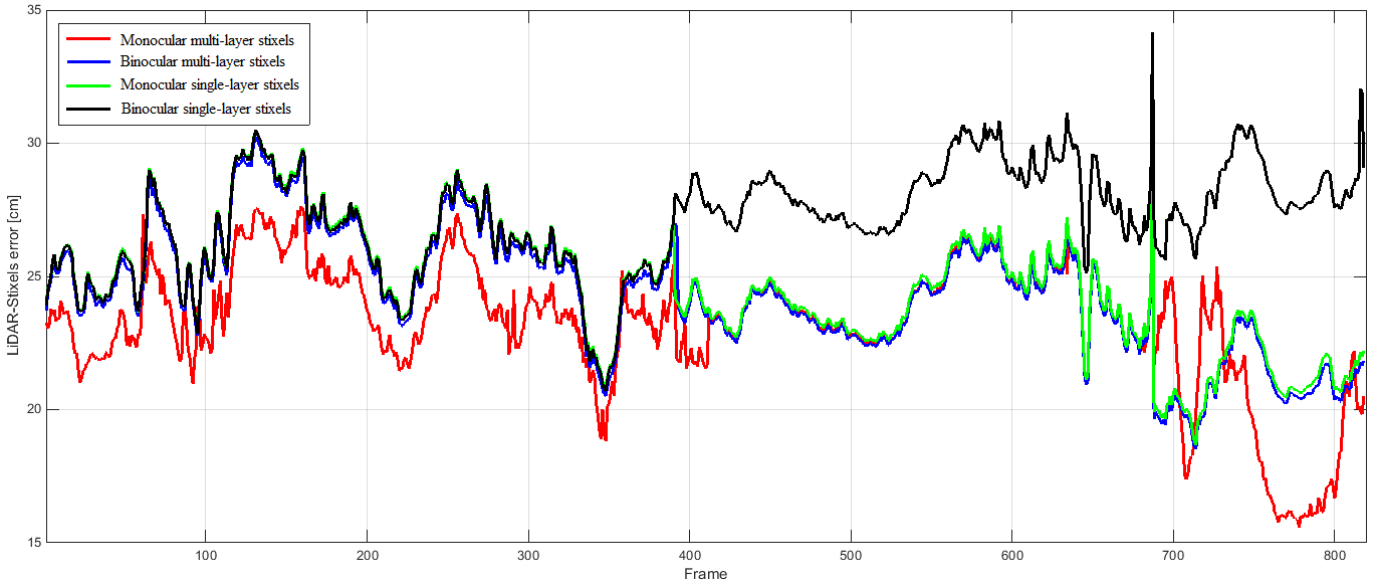

Fig. 5. Error rates illustrate the mean of LIDAR-stixel distance error (in cm) for the four approaches. The frame ranges [1-390], [391-688], and [689-820] represent categories B, A and C, respectively.

for each column x where $\mathbb{L}$ is a set of possible segmentations. A segment s_x^n is represented by

$$s_x^n = \{y_n^b, y_n^t, c_n, f_n(\cdot)\} \tag{19}$$

with $1 \leq y_n^t \leq y_n^b \leq N_{row}$ (note: y-coordinates go downward; thus the top coordinate y_n^t is less than or equal to the bottom coordinate y_n^b), $c_n \in \mathbb{C}$, and the function $f_n(\cdot)$ is defined for y, with $y_n^t \leq y \leq y_n^b$, for the disparity of segment s_n at row y. Functions $f_n(\cdot)$ satisfy the following properties:

- Ground-based stixels are generated based on a ground disparity map GD. This enables us to determine the *ground function* $f_g(\cdot)$ (note: "g" instead of number $n = 1$) and identify the road surface using a single camera after resolving the displacement issue.
- For an *object function*, we have that $f_o(y) = \mu_n$ (note: "o" instead of a number n between 1 and N_x) where μ_n is the mean disparity within s_n. We extend this function to enable transitivity error analysis to be used for valid disparity coverage.

Moreover, the objective of $f_n(y)$ is to compute the disparity of that segment s_n at row y. This step will arise as a typical *maximum-a-posteriori* (MAP) estimation problem. We will find the most probable labelling and solve

$$L^* = \text{argmax}_{L \in \mathbb{L}} P(L|D) \tag{20}$$

where $L \in \mathbb{L}$ is an ordered list of N_x adjacent and non-overlapping stixel segments s_n.

IV. Experimental Results

The accuracy of the proposed monocular stixel method was evaluated and compared to the results of the original binocular base-line stixel method. The KITTI dataset [14], that includes a diverse collection of traffic scenes, was used for the experiment. In total, 818 stereo images were tested in the *road* sequence (category A, and B), and *residential* (category C) which are all available in the KITTI datasets (description provided in Table I). We aimed at having a wide diversity of challenging traffic situations including low-textured roads, different road views, and challenging obstacle surfaces. For evaluation purposes, stixel-LIDAR depth was used as ground truth, as suggested in [11]. All stixels, in every frame, were evaluated individually based on processes as discussed in [11].

TABLE I
SELECTED TEST SEQUENCES FROM THE KITTI DATASET.

Category	Sequence	Frames
A	2011_09_26_drive_0015	297
B	2011_09_26_drive_0032	390
C	2011_09_26_drive_0035	131

Mean distance differences are summarized in Table II. Error rates are plotted in Fig. 5 for road and residential data. The number of errors is highly reduced when using the proposed monocular stixel approach. Figure 6, for example, illustrates the accuracy of the proposed monocular+LIDAR method (multi-layer and single-layer) for challenging obstacle detection conditions. Resulting stixels, using monocular+LIDAR multi-layer, are more accurate than the original binocular ones. By visual evaluation, the original method has some limitations in identifying road surfaces and objects independently in

TABLE II
LIDAR-BASED QUANTITATIVE EVALUATION BASED ON MEAN DISTANCE ERROR [CM] OF STIXELS USING KITTI DATASET (BINOCULAR AND MONOCULAR CONFIGURATION).

Sequence	Single layer stixels				Multi-layer stixels			
	Binocular		Monocular		Binocular		Monocular	
	μ	σ	μ	σ	μ	σ	μ	σ
A	2.82	1.25	2.42	1.25	2.40	1.25	2.39	1.35
B	2.60	1.95	2.61	1.94	2.58	1.93	2.38	1.76
C	2.80	1.31	2.12	1.15	2.98	1.14	1.96	2.95

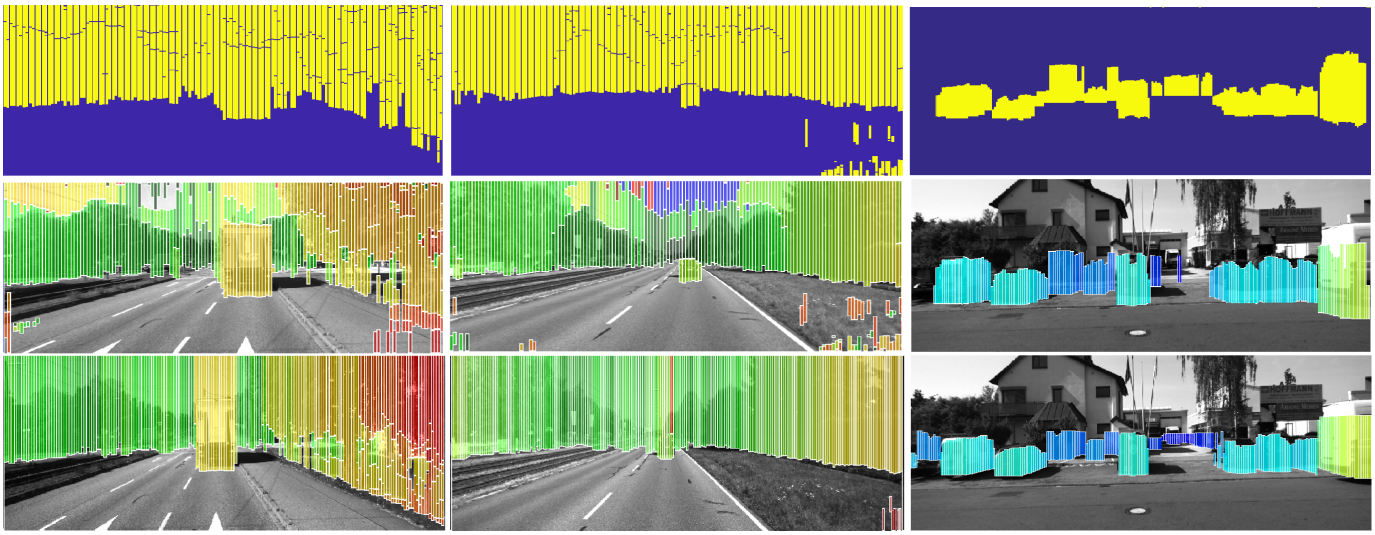

Fig. 6. Qualitative results. *Top row*: Stixel maps. *Middle row*: Estimation using a disparity map resulting from stereo matching (binocular). *Bottom row*: Estimation using a disparity map incorporating LIDAR data (monocular).

TABLE III
RUN-TIME PROFILING FOR MULTI-LAYER STIXELS ON KITTI DATASET.

Category	Base-line binocular	Proposed monocular+LIDAR
A	27.3 s	23.3 s
B	26.9 s	22.6 s
C	25.7 s	22.7 s

the disparity map. This problem occurred several times in tested categories. Thus, the LIDAR points play an essential role in providing an accurate disparity map, and their use also minimises the processing time required to generate the disparity map (see Table III). This indicates an optimised balance between disparity-map accuracy and processing time in terms of stixel estimation. The limited number of LIDAR points acquired by the sensor define a limitation in our method. As we can observe from Fig. 5, the distance error in (category A) between proposed monocular and conventional binocular multi-layer was very close; this also occurred in open-road scenarios with shadows.

V. CONCLUSIONS

This paper proposed an approach for robust stixel detection using monocular vision and LIDAR data. The main benefit of our work is to optimise the balance between accuracy and processing time when generating stixels. The proposed method has been compared to the original base-line method. Experiments show that the error rate was reduced (by 15.4 % on tested data) using monocular+LIDAR stixels. Results demonstrate the potential of this novel method towards more accurate obstacle surface detection and recovering of low-textured road information.

REFERENCES

[1] S. Edelstein. Velodyne just cut the price of its most popular Lidar sensor in half. www.thedrive.com/tech/17297/, retrieved September 01, 2018.

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

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

[4] D. Pfeiffer. The stixel world. Doctoral thesis, Humboldt University Berlin, 2011.

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

[6] R. Benenson, M. Mathias, R. Timofte, and L. Van Gool. Fast stixels estimation for fast pedestrian detection. In Proc. *European Conf. Computer Vision*, 11–20, 2012.

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

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

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

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

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

[12] D. Hernandez, A. Espinosa, J. Moure, D. Vázquez, and A. López. GPU-accelerated real-time stixel computation. In Proc. *Winter Conf. Applications Computer Vision*, 1054–1062, 2016.

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

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

[15] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The KITTI dataset. *Int. J. Robotics Research*, 32:11, 1231–1237, 2013.

[16] C. Premebida, L. Garrote, A. Asvadi, A. Pedro Ribeiro, and U. Nunes. High-resolution LIDAR-based depth mapping using bilateral filter. In Proc. *Computer Vision Pattern Recognition*, 2016.

[17] J. Dolson, J. Baek, C. Plagemann, and S. Thrun. Upsampling range data in dynamic environments. In Proc. *Computer Vision Pattern Recognition*, 2010.

[18] N. H. Saleem, H.-J. Chien, and R. Klette. Stixel optimization: Representing challenging on-road scenes. In Proc. *Int. Conf. Image Vision Computing New Zealand*, IEEE *Xplore*, 2017.