2014 IEEE 17th International Conference on
Intelligent Transportation Systems (ITSC)
October 8-11, 2014. Qingdao, China

Modeling Arbitrarily Oriented Slanted Planes for Efficient Stereo Vision based on Block Matching

Benjamin Ranft and Tobias Strauß

***Abstract*— Stereo cameras enable a 3D reconstruction of viewed scenes and are therefore well-suited sensors for many advanced driver assistance systems and autonomous driving. Modern algorithms for estimating distances for every image pixel achieve high-quality results, but their real-time capability is very limited. In contrast, window-based local methods can be implemented very efficiently but are more prone to errors. This is particularly true for spatial changes of distance within the matching window, most prominently on surfaces such as the road which are not parallel to but rather slanted towards the image plane. In this paper we present a method to compensate the impact of this effect for arbitrarily oriented sets of planes. It does not depend on any modifications to the actual distance estimation. Instead, it only applies specific transformations to input images and intermediate results. By combining this approach with existing implementations which efficiently use either multi-core or graphics processors, we were able to significantly increase quality while maintaining real-time throughputs on a compact target system.**

I. INTRODUCTION

Many advanced driver assistance systems and autonomous driving depend on the perception of the static environment and dynamically moving objects. Such information is extracted mainly from measurements of 3D sensors. For this purpose, e. g. the autonomous vehicle navigating the Bertha Benz Memorial Route in 2013 [1] employed a stereo camera. This type of sensor commonly consists of two side-by-side cameras observing the same scene. 3D distances can be reconstructed from its left and right 2D images by finding the horizontal offset for which the images' content has the most similar appearance. This offset is called disparity and is inversely proportional to an object's distance. To avoid ambiguities, usually the similarity of rectangular windows rather than single pixels is optimized. Our disparity estimation presented in [2] can achieve latencies below 12 ms for images sized 0.5 Mpx using standard processors. However, like virtually all window-based methods, it applies a constant disparity across the matching window and consequently assumes that the scene consists of frontally viewed planes which are perpendicular to the camera's optical axis. In the context of ground vehicles, this assumption often does not hold – not only on the road surface but also on roadside objects such as building facades. As shown exemplarily in fig. 1, this can lead to inaccurate, incomplete or incorrect results. Fig. 2 gives a preview that the method to be presented enables a significant reduction of these effects. Additionally, our method provides a classification of surface orientation as a by-product. It is not limited to the specific cases of planes horizontal or vertical w. r. t. the camera, and is therefore also well-suited for the significant roll angles of motorcycles and other two-wheelers.

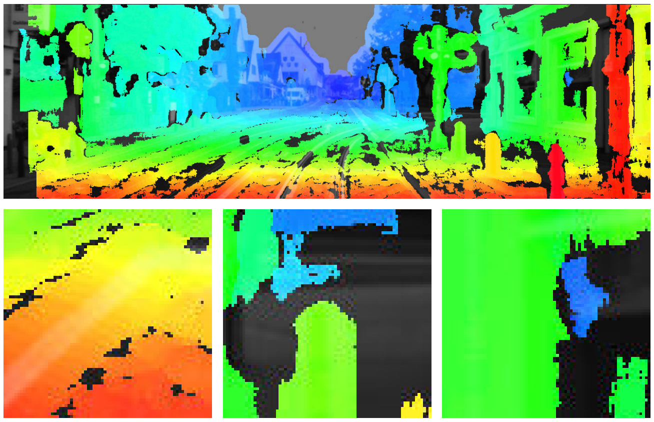

Fig. 1. Result of a typical local method for disparity estimation, magnified inaccurate (left), incomplete- (center) and incorrect regions (right)

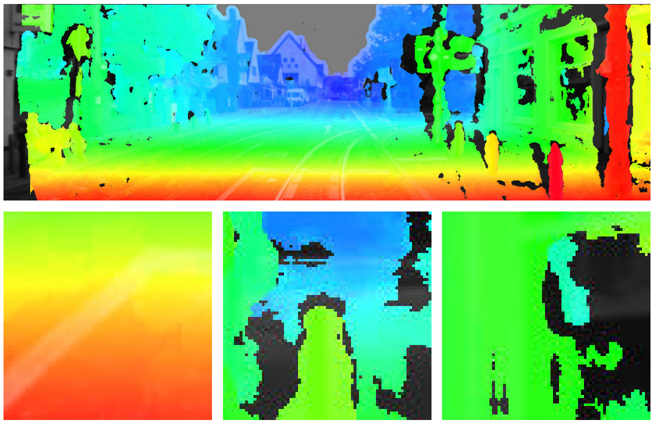

Fig. 2. Result of the proposed method after explicitly modeling road and building facade as slanted planes, same magnified regions as in fig. 1

The corresponding approach constitutes the focus of this paper in section IV. In advance, section II presents related work and explains differences from this paper's contributions. Our baseline implementation for common frontal stereo vision is presented in section III, including a variant of the Census Transform which increases robustness while remaining efficiently computable and comparable. Section V shows results of a quantitative evaluation of the quality and performance of our and other methods, while section VI concludes the paper and gives an outlook to future works.

B. Ranft is with FZI Research Center for Information Technology, 76131 Karlsruhe, Germany ranft@fzi.de

T. Strauß is with the Department of Measurement and Control Systems, Karlsruhe Institute of Technology (KIT), 76131 Karlsruhe, Germany strauss@kit.edu

978-1-4799-6078-1/14/$31.00 ©2014 IEEE

II. RELATED WORK

To increase the quality of disparity estimates, i. a. [3] uses the Census transform which describes a pixel as a bit-string: One bit each encodes whether the pixel is brighter than a specific neighbor. Two bits each allow an additional comparison to the mean of all evaluated neighbors [4] or – for the sake of robustness – a classification into clearly darker, similar and clearly brighter neighbors [5]. The dissimilarity of two pixels can be quantified by their *Hamming* distance, i. e. the number of different bits in their respective descriptors. Our algorithm for efficiently computing this metric for several pairs of pixels is a further optimization of [3].

A variety of approaches is dedicated to planes viewed at a slant, particularly to the special case of a horizontal ground plane at a negligible roll angle: [6] attempts to map a statically defined ground plane from the right to the left image with a fixed horizontal shear and shift; an exhaustive disparity search is conducted only where that attempt has failed. In contrast, [9] processes vertical surfaces such as walls by comparing single image row segments not only at different disparities but also after multiple horizontal scalings. [7] applies a local disparity estimation method to original as well as statically sheared images; a fusion of both disparity maps achieves improved results on the road surface at a run-time of 430 ms. Similarly, our proposed method estimates disparities from differently transformed input images. However, by combining shearing and scaling, it is able to ideally handle a superset of slanted planes and to select hypotheses dynamically based on the current scene. A dynamic adaptation of only the shear gradient allows [8] to detect obstacles without explicitly computing disparities.

[10] proposed an extension to the well-known semi-global matching for handling previously classified horizontal or vertical surfaces: Reducing the specific costs caused by their respective orientation class increased result quality but also run-time. The method presented in [11] can ideally process arbitrary planes but practically assumes a “Manhattan world”: The hypotheses consist of three orthogonal stacks of planes – frontal, horizontal and vertical. It is real-time capable at a resolution of 512 x 384 px and on a high-end GPU. Section V will show that our implementation lowers the requirements to either a midrange consumer CPU or GPU even at 2.5 times the resolution. Dynamically handling many arbitrarily-oriented planes is reserved to complex methods e. g. based on super-pixels and Markov networks: [12] has been able to significantly improve their efficiency, but still requires between several seconds and few minutes per frame.

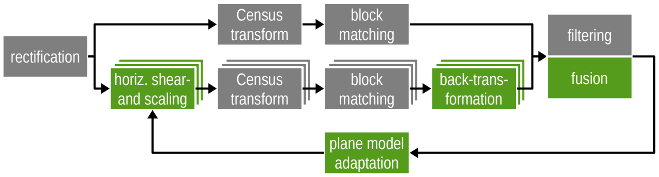

Fig. 3. Estimation of standard disparities (upper path) and multiple pseudo-disparities on differently oriented slanted plane sets (middle paths), adaptation of plane hypotheses based on previous results (lower feedback path): new plane model-specific (green) and generic processing steps (gray)

Supplementing disparity estimation with a separate [13] or integrated [14][15] estimation of optical flow enables the reconstruction of not only 3D positions but also their velocity vectors – a valuable information for advanced driver assistance and autonomous driving. The implementation described in the following is generally suitable for flow estimation as well, but we have not yet pursued this application.

III. BASELINE IMPLEMENTATION

As a basis for disparity estimation on slanted planes, we would like to summarize relevant properties and further developments of our initial standard method. In [2] we developed and compared implementations for either multi-core, graphics or embedded processors. Built upon this, [16] presents an adaptive cooperation of CPUs and GPUs. Our method’s core is classical block matching, which quantifies the dissimilarity of windows in the left and right image by their respective pixels’ sum of absolute differences (SAD). By using running sum tables, this metric can be computed with two additions and subtractions each, independently from window size and in a cache-friendly way. Additionally, the SIMD vector instructions of modern CPUs enable concurrent evaluation of 16 disparities per core.

Fig. 3 shows our method’s complete processing chain with steps pre- and succeeding the aforementioned block matching. A Census transform variant has replaced the previous difference-of-Gaussians high-pass filter. The general goal of such preprocessing is to make the similarity between left and right image windows robustly and quickly measurable i. a. by mitigating illumination differences. In contrast to the high-pass filter, the Census transform can not only compensate for additive but also multiplicative offsets. For each pixel, our variant computes a 16-bit descriptor by concatenating 2 bits for each of 8 neighbors at a distance of $s = 4$ px:

$$C\left(u,v\right) = \bigotimes_{i,j=-1}^{i,j=1} \begin{cases} 00, & I(u,v) - t \geq I(u{+}si, v{+}sj) \\ 01, & I(u,v) - t < I(u{+}si, v{+}sj) \wedge \\ & I(u,v) + t \geq I(u{+}si, v{+}sj) \\ 11, & I(u,v) + t < I(u{+}si, v{+}sj) \end{cases} \tag{1}$$

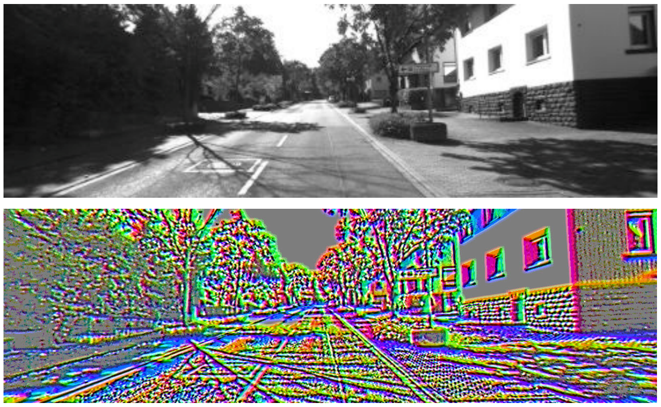

Fig. 4. Input image example and visualization of its 2-bit/neighbor Census transform: The blue/green/red channel encodes a center similarly bright or (at full saturation) brighter than upper/lower-left/lower-right neighbor pixels.

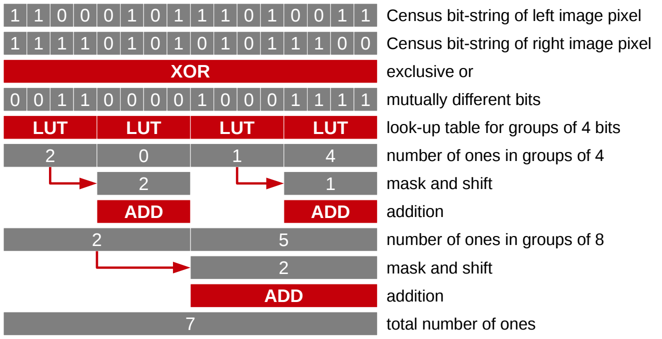

Fig. 5. Sample *Hamming* distance computation of two Census descriptors, applicable in parallel to 15 more descriptor pairs per CPU core (not shown)

The brightness tolerance $t = 2$ creates an interval of gray values considered equal to the central pixel, which reduces susceptibility to noise. The binary encoding is defined such that the *Hamming* distance between this "equally bright" relation and a significantly brighter or darker neighbor pixel is 1, while the distance between "brighter" and "darker" is 2. Fig. 4 shows a visualization of this Census transform. Our CPU and GPU implementations thereof reduced preprocessing run-time w. r. t. the previous high-pass filter by 70 % each.

A pixel's Census descriptor is computed only once, but compared to another for each tested disparity. Therefore it is crucial that the *Hamming* distance of 16 bits each can be computed efficiently. The XOR instruction quickly marks identical bits with 0 and different bits with 1; the latter must then be counted. Some processors offer native instructions to find this "population count". Alternatively, a look-up table with $2^{16} = 65536$ entries achieves a similar throughput for one pixel at a time. However, since our block matching implementation evaluates 16 disparities concurrently, the same number of *Hamming* distances must be computed at once. Fig. 5 outlines an algorithm for this purpose which only uses SIMD-capable instructions and achieves six times the throughput of the aforementioned variants. It is based on [3], which counts 1-bits within groups of 2, 4, 8 and finally 16 bits. To further improve performance we replaced the first two stages by a LUT within a processor register, which directly provides the count of set bits within groups of four.

IV. DISPARITY ESTIMATION ON SLANTED PLANES

As mentioned initially, block matching on non-frontal surfaces is complicated by the fact that their distance to the camera is not constant across the matching window. Nevertheless, every plane in world coordinates (X, Y, Z) also corresponds to a plane in the parameter space of image column u, row v and disparity d, so its gradients $\partial d/\partial u$ and $\partial d/\partial v$ are constant. Consequently, the desired constant pseudo-disparity d_t between left and right image can be achieved by horizontally shearing and scaling one or both images with suitable parameters. As a preview, fig. 6 shows that – after a strict filtering based on the sums of *Hamming* distances across the matching window – mostly disparities on either vertical structures or the ground remain respectively. Since the left image is commonly used as a reference, we apply such transformations to the right image only: Its pixels are horizontally shifted based on the shear gradient g and the scale factor s. No vertical shift is required, so a key property of rectified images – all corresponding points being located in the same image row – is not affected by our method.

$$u_t = s\left(u_r + gv\right) \tag{2}$$

$$\begin{aligned} d_t &= u_l - u_t \\ &= s\left(u_l - u_r\right) + \left(1 - s\right) u_l - sgv \\ &= sd + \left(1 - s\right) u_l - sgv \end{aligned} \tag{3}$$

We refer to the offset between original left and transformed right image d_t as pseudo-disparity because it is estimated by the same methods as standard disparities but must be interpreted differently. The following derivation of the transformation parameters g and s also explains this interpretation.

Fig. 7 shows a stereo camera with baseline width B. It observes a plane at a roll angle Φ which is slanted towards the camera's optical axis by Θ. The distance D between the plane and the left camera's optical center is measured within the XY-plane. The following equation holds for every point (X, Y, Z) on the plane:

$$D = X\sin\left(\Phi\right) + Y\cos\left(\Phi\right) + Z\tan\left(\Theta\right) \tag{4}$$

Inserting $X = Z\left(u_l - u_0\right)/f$, $Y = Z\left(v - v_0\right)/f$ and $Z = Bf/d$ yields a relation to the disparity d, given the camera's focal length f and principal point (u_0, v_0):

$$\begin{aligned} d = \frac{B}{D}\Big(& f\tan\left(\Theta\right) \\ &+ \left(u_l - u_0\right)\sin\left(\Phi\right) \\ &+ \left(v - v_0\right)\cos\left(\Phi\right)\Big) \end{aligned} \tag{5}$$

The tuple of shear gradient g and scale factor s which yields the desired constant pseudo-disparity on the given plane can now be found from the partial derivatives by image row and column of (3):

$$\begin{aligned} \frac{\partial d_t}{\partial v} &= s\left(\frac{B}{D}\cos\left(\Phi\right) - g\right) \overset{!}{=} 0 \\ \rightarrow \quad g &= \frac{B}{D}\cos\left(\Phi\right) \end{aligned} \tag{6}$$

$$\begin{aligned} \frac{\partial d_t}{\partial u_l} &= 1 + s\left(\frac{B}{D}\sin\left(\Phi\right) - 1\right) \overset{!}{=} 0 \\ \rightarrow \quad s &= \frac{D}{D - B\sin\left(\Phi\right)} \end{aligned} \tag{7}$$

The transformation parameters are not affected by the pitch or yaw angle[1] Θ between plane and optical axis. Instead, as shown in fig. 8, an interval of discrete pseudo-disparities d_t

[1] The suitable name depends on the roll angle between plane and camera.

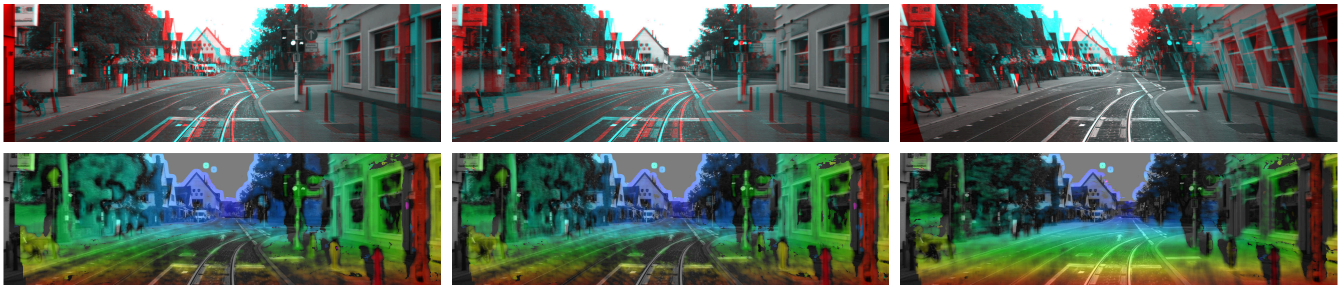

Fig. 6. Superimposed input images (top) and intermediate disparity maps from single sets of plane hypotheses (bottom): aligned traffic light and sign from standard estimation (left), aligned building facade after scaling the right image (center), aligned road surface after shearing the right image (right)

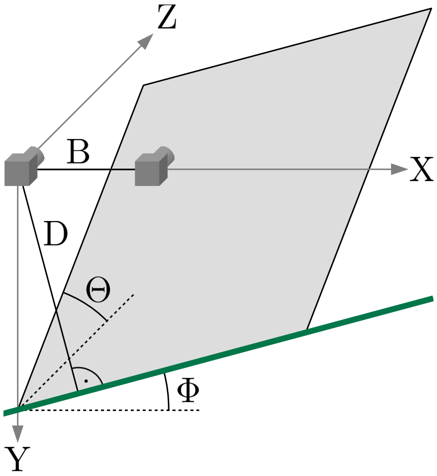

Fig. 7. Stereo camera with baseline width B, plane slanted by Θ towards the camera at roll angle Φ and distance D within the image plane

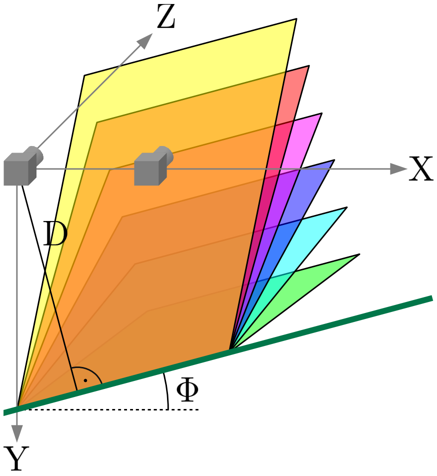

Fig. 8. Set of slanted plane hypotheses modeled by an interval of pseudo-disparities d_t for a single right image transformation (g, s)

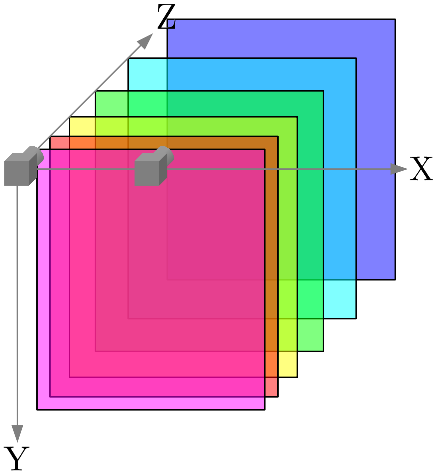

Fig. 9. Stack of frontal plane hypotheses modeled when evaluating an interval of standard disparities d on original left and right images

forms a set of planes observed at different angles Θ whose intersection is a straight line within the XY-plane:

$$d_t = \frac{B}{D - B\sin(\Phi)}\Big(f\tan(\Theta) - u_0\sin(\Phi) - v_0\cos(\Phi)\Big) \quad (8)$$

In contrast and for comparison, fig. 9 shows the stack of frontally viewed planes which are modeled and evaluated by a standard block matching method.

As the processing chain in fig. 3 already indicated, multiple image transformations and pseudo-disparity estimations may be conducted in order to evaluate differently oriented sets of slanted planes, e. g. one set for the ground at different slopes and another set for roadside buildings at different yaw angles. We propose an adaptive method for determining their number as well as their respective transformation parameters: Without supplementary information, tuples (g, s) adapted to the current scene can be obtained from an already available disparity image – either the previous fused result of an image sequence or a frontal-only estimation based on the current image pair. The desired tuples appear as local maxima in a 2D histogram of the disparity image's spatial gradients according to:

$$g = \frac{\partial d}{\partial v} \qquad s = \left(1 - \frac{\partial d}{\partial u_l}\right)^{-1} \quad (9)$$

To obtain more precise values of (g, s), the discretely quantized disparity images are down-sampled and their gradients are measured across several pixels. This also contributes to this approach's low latency of only 0,1 ms, which is significantly less than the standard approach of robustly fitting a plane into a 3D point cloud with a RANSAC scheme. Fig. 10 shows examples of such histograms as well as corresponding disparity maps.

Once the right input image has been sheared, scaled and Census transformed, the pseudo-disparities between it and the left image may be estimated using any existing implementation. Nevertheless, it should be noted that constant disparity search range limits $[d_{min}, d_{max}]$ are mapped to an interval of d_t which depends on the pixel coordinates (u, v) according to (3). However, our estimation method [2] draws its performance i. a. from applying the same (pseudo-)disparity interval to a rectangular region of interest. Consequently, this interval needs to be the union of each included pixel's interval. This is why the estimation of pseudo-disparities takes about 10 % longer than that of

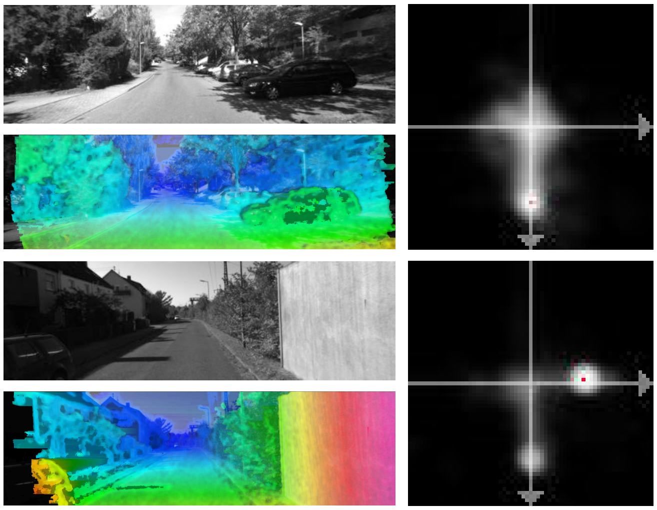

Fig. 10. Left input images and disparity maps (left), histograms of disparity gradients $\partial d/\partial u$ and $\partial d/\partial v$: Their indicated maximum represents the road surface (top) and the right hand side wall (bottom) respectively.

regular disparities[2]. Our GPU implementation processes tiles sized 54 x 64 px independently. A tile's minimum pseudo-disparity to be evaluated can be derived from its position in the left camera image $[u_{l,min}, u_{l,max}]$ x $[v_{min}, v_{max}]$:

$$\begin{aligned} d_{t,min} = s\, d_{min} + (1-s) &\left\{ \begin{array}{ll} u_{l,min}, & s < 1 \\ u_{l,max}, & s \geq 1 \end{array} \right. \\ - s\, g &\left\{ \begin{array}{ll} v_{min}, & g < 0 \\ v_{max}, & g \geq 0 \end{array} \right. \end{aligned} \tag{10}$$

The corresponding maximum pseudo-disparity $d_{t,max}$ is found analogously by interchanging the min and max subscripts above. In contrast, our multi-core CPU implementation assigns to each core a horizontal stripe of the result disparity image which spans the image's full width. To still avoid unnecessary computations, each SIMD group of 16 (pseudo-)disparities $[d_{t,min}, d_{t,max}]$ is only applied to an individual column range $[u_{l,min}, u_{l,max}$ (again analogously)] according to:

$$\begin{aligned} u_{l,min} = \frac{1}{1-s} \Bigg(&\left\{ \begin{array}{ll} d_{t,min} - s\, d_{max}, & s < 1 \\ d_{t,max} - s\, d_{min}, & s \geq 1 \end{array} \right. \\ & - s\, g \left\{ \begin{array}{ll} v_{min}, & (1-s)\, g < 0 \\ v_{max}, & (1-s)\, g \geq 0 \end{array} \right. \Bigg) \end{aligned} \tag{11}$$

After each estimation of (pseudo-)disparities we filter the results as already presented in [2] and [16]: For each pixel, both its optimal sum of *Hamming* distances and the consistency of its disparities w. r. t. left and right image are converted into a confidence metric $\in [0, 256)$ using a manually parameterized fuzzy logic network. For this purpose, the intermediate results from each sheared and scaled image must be transformed back to right image columns u_r and regular disparities d according to (3):

$$u_r = \frac{u_t}{s} - gv \tag{12}$$

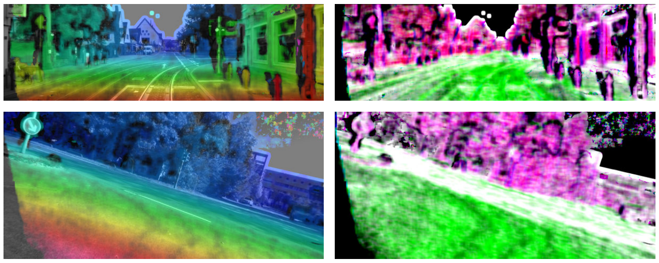

Fig. 11. Fused disparity (left) and individual confidence images (right) for different plane model hypotheses: The green/blue/red channel encodes the confidence of the best horizontal/vertical/frontal plane respectively.

$$\begin{aligned} d &= \frac{d_t}{s} + \left(1 - \frac{1}{s}\right) u_l + gv \\ &= d_t\, (s-1)\, u_r + sgv \end{aligned} \tag{13}$$

For the final fusion of one regular and multiple pseudo-disparity estimates from differently oriented sets of slanted planes, an approach similar to [7] has proven useful: The plane hypothesis with the lowest sum of *Hamming* distances metric is selected independently for each pixel. However, only result candidates with a non-zero confidence – i. e. those which also passed the left-right consistency check – are taken into account.

Fusion as the last processing step also offers the initially mentioned opportunity to perform a classification of surface orientations within the viewed scene. If e. g. the confidence values from a transformation $(g > 0.2, s \approx 1)$ – typical parameters for ground planes – are significantly higher than those from standard block matching and other transformations, this is a strong indicator for the road surface. Particularly the latter is clearly marked by green areas of fig. 11, while a distinction between vertical structures viewed either frontally or at a slant cannot be made as easily, as their predominant magenta hue shows. The figure also presents a sample result from a data set obtained with a motorcycle. However, due to the lack of corresponding ground truth data, the following evaluation cannot focus on two-wheelers in more detail.

V. EXPERIMENTAL EVALUATION

The following section includes a quantitative evaluation of the presented method w. r. t. two criteria: run-time and result quality. For the interpretation of the former, we would like to briefly introduce our target system which requires significantly less power and space than the workstation used in [2][16]. It contains one midrange CPU and GPU each:

- The *Core i7-4771* CPU offers four cores at 3.5 GHz and supports the new vector instruction set *AVX2*, which allows our implementation to concurrently evaluate 16 rather than 8 pixels at a time. It consumes at most 84 W.
- The *GeForce GTX 760* GPU uses up to 170 W to run 1152 cores at 1 GHz. In spite of the lack of vector instructions, our implementation processes 2 pixels/core.

Our input data comes from the *KITTI Vision Benchmark Suite* [17], whose rectified stereo image pairs are about 0.5 Mpx in size and have been recorded with a prototype car in a

[2] This value would likely be higher for (semi-)global rather than local methods since they cannot be applied independently to image partitions.

TABLE I

CRITERIA FOR EACH VARIANT OF THE PROPOSED METHOD FROM THE *KITTI Stereo Evaluation* [17]: RATIO OF DISPARITIES WITH ERRORS OF AT LEAST 3 PX ("OUT") AND AVERAGE ERROR ("AVG"), WITHOUT ("NOC") AND INCLUDING ("ALL") PIXELS OCCLUDED FOR THE RIGHT CAMERA

preprocessing	plane model	"Out-Noc"	"Out-All"	"Avg-Noc"	"Avg-All"	density	run-time
high-pass	frontal only	14.27 %	15.57 %	2.5 px	2.8 px	77 %	12.0 / 8.7 ms
Census	frontal only	8.36 %	9.72 %	1.8 px	2.1 px	84 %	11.8 / 8.5 ms
high-pass	additional slanted	12.62 %	13.97 %	2.1 px	2.5 px	81 %	32.9 / 23.0 ms
Census	additional slanted	6.16 %	7.42 %	1.2 px	1.4 px	87 %	32.5 / 22.6 ms

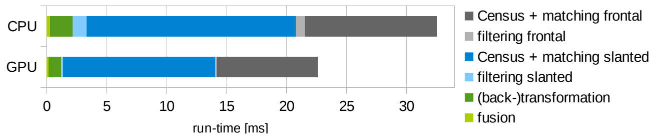

Fig. 12. Breakdown of run-times of the processing steps involved in the proposed method: With only 0.05 ms/frame, the Census transform would have been indistinguishable if listed individually.

mostly urban environment. Therefore, the camera's roll angle Φ w. r. t. the ground or roadside walls always nearly equals 0 or $\pm\pi/2$ respectively. Consequently, our methods capability of modeling and processing differently oriented planes is only used occasionally. Nevertheless, the histogram-based approach for adapting the plane models to the current scene has proven helpful: On average, 1.43 transformations and plane sets were processed in addition to standard frontal matching. The road surface was explicitly modeled in 96 % of all images and only omitted if its view was obstructed. The upper limit of three additional models e. g. for handling the road as well as left and right roadside walls was reached in 7.8 % of cases.

Based on the above numbers one can expect that the run-time of the proposed method almost tripled w. r. t. the initial frontal planes-only implementation. Both the last column of table I and the detailed breakdown of latencies per processing step in fig. 12 confirm this expectation. Nevertheless, throughputs of 31 or 44 Hz on CPU or GPU respectively still allow real-time operation. Concerning power efficiency in the form of throughput per watt, our implementations and test system achieve 0.37 (CPU) or 0.26 Hz/W (GPU). For comparison, we estimated 0.55 Hz/W (1.50 Hz/W if only frontal) for a modern implementation [18] based on an FPGA rather than general-purpose processors when applied to the same input data.

The evaluation of our method's result quality via the *KITTI Stereo Evaluation* is based on comparisons with ground truth data obtained from a *Velodyne HDL-64E* laser scanner. Table I summarizes corresponding statistics of all described variants of our method: Both our version of the Census transform and the adaptive modeling of slanted planes have contributed significantly to the improved result quality. At the same time, the Census transform even slightly improved throughput as well. Conclusively, fig. 13 presents a quantitative comparison of our best variant with the other participants in the *KITTI Stereo Evaluation*: While we achieved a midrange position in the quality ranking, our implementation's computational performance is faster by one order of magnitude w. r. t. methods with similar quality and by two orders of magnitude when compared to algorithms yielding a significantly higher quality. Both statements reflect the ranking's state as of August 2014. Its most recent version is available at `http://www.cvlibs.net/datasets/kitti/eval_stereo_flow.php`.

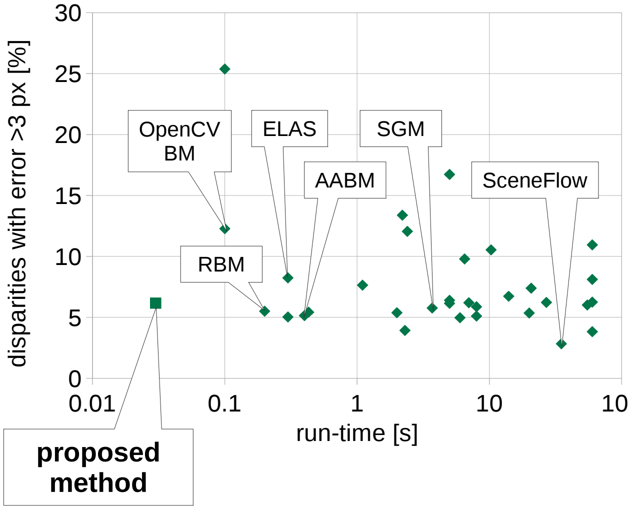

Fig. 13. Comparison of both result quality and run-time of the proposed method with the other participants in the *KITTI Stereo Evaluation* [17]

VI. CONCLUSIONS AND FUTURE WORKS

We have presented further developments of a local method for disparity estimation which is real-time capable due to its algorithmic efficiency and parallel processing. A first increase in result quality has been achieved with a new variant of the Census transform for image preprocessing, whose per-pixel descriptors are efficiently computable and comparable. The focus of this paper however is the algorithm for ideally processing slanted planes with arbitrary orientations based on fast and simple block matching methods. This capability is not only but particularly relevant to two-wheeled vehicles where significant roll angles occur. To conclude, fig. 14 presents the effects of these two improvements once more in the form of 3D point clouds reconstructed by each variant. The run-time of our implementation grows nearly linearly with each evaluated set of slanted planes, but it is nevertheless much faster than that of other methods with comparable result quality. We have achieved real-time throughputs on midrange standard CPUs as well as GPUs, and narrowed the gap towards FPGAs in terms of energy efficiency.

Future development will focus on the filter and fusion processing steps. The fuzzy logic for determining the confidence of each estimated disparity and the independent per-pixel selection of the fused plane hypothesis will be replaced by a trained probabilistic model. This might allow obtaining sufficiently good results even without checking the consistency of disparities w. r. t. left and right image, so that the time-consuming estimation of the latter may be omitted. Besides that, we will apply the strategies for automatically adapting parallel processing presented in [19] to this more complex disparity estimation method and concurrently-running autonomous driving applications which use its results.

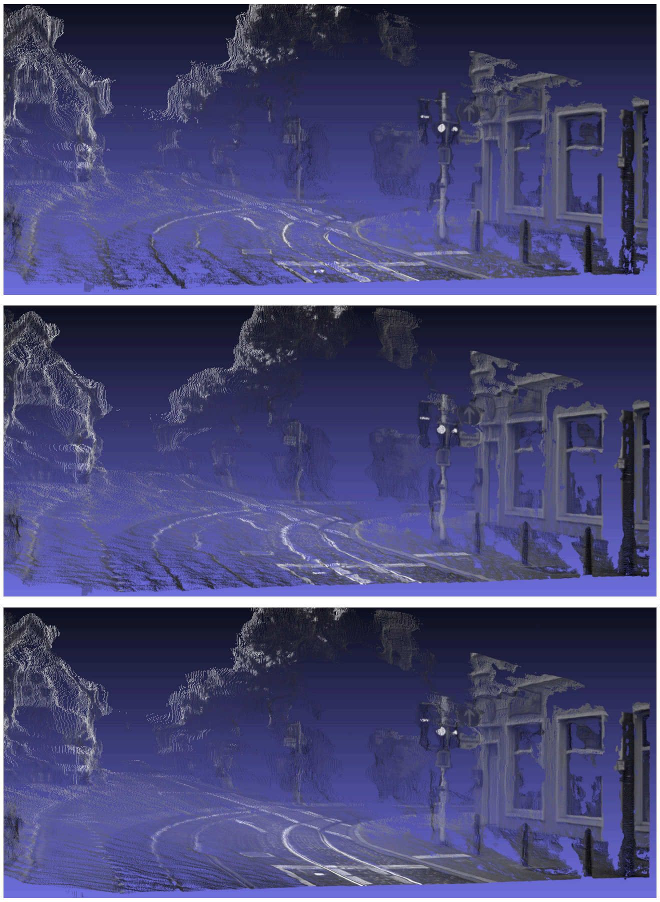

Fig. 14. Reconstructed 3D point clouds from the initial method (top), after replacing high-pass preprocessing with our variant of the Census transform (center), and after modeling not only frontal but also slanted planes (bottom)

References

[1] U. Franke, D. Pfeiffer, C. Rabe, C. Knoeppel, M. Enzweiler, F. Stein, R. G. Herrtwich, ”Making Bertha See”, *in IEEE International Conference on Computer Vision, Workshop Computer Vision for Autonomous Vehicles*, 2013.

[2] B. Ranft, T. Schönwald and B. Kitt, ”Parallel Matching-based Estimation – a Case Study on Three Different Hardware Architectures”, *in IEEE Intelligent Vehicles Symposium*, 2011.

[3] C. Zinner, M. Humenberger, K. Ambrosch and W. Kubinger, ”An Optimized Software-Based Implementation of a Census-Based Stereo Matching Algorithm”, *in International Symposium on Visual Computing*, 2008.

[4] L. Ma, J. Li and H. Zhang, ”A Modified Census Transform Based on the Neighborhood Information for Stereo Matching Algorithm”, *in IEEE International Conference on Image and Graphics*, 2013.

[5] F. Stein, ”Efficient Computation of Optical Flow Using the Census Transform”, *in DAGM Pattern Recognition Symposium*, 2004.

[6] P. Burt, L. Wixson and G. Salgian, ”Electronically Directed “Focal” Stereo”, *in IEEE International Conference on Computer Vision*, 1995.

[7] N. Einecke and J. Eggert, ”Stereo Image Warping for Improved Depth Estimation of Road Surfaces”, *in IEEE Intelligent Vehicles Symposium*, 2013.

[8] K. H. Won and S. K. Jung, ”Ground Plane Stereo for Obstacle Detection”, *in International Conference on Image and Vision Computing New Zealand*, 2011.

[9] A. S. Ogale and Y. Aloimonos, ”Stereo Correspondence with slanted surfaces: critical implications of horizontal slant”, *in IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[10] R. Spangenberg, T. Langner and R. Rojas, ”Weighted Semi-Global Matching and Center-Symmetric Census Transform for Robust Driver Assistance”, *in International Conference on Computer Analysis of Images and Patterns*, 2013.

[11] D. Gallup, J.-M. Frahm, P. Mordohai, Q. Yang and M. Pollefeys, ”Real-Time Plane-Sweeping Stereo with Multiple Sweeping Directions”, *in IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[12] K. Yamaguchi, T. Hazan, D. McAllester and R. Urtasun, ”Continuous Markov Random Fields for Robust Stereo Estimation”, *in European Conference on Computer Vision*, 2012.

[13] C. Rabe, T. Mller, A. Wedel and U. Franke, ”Dense, Robust, and Accurate Motion Field Estimation from Stereo Image Sequences in Real-time”, *in European Conference on Computer Vision*, 2010.

[14] J. Cech, J. Sanchez-Riera and R. Horaud, ”Scene Flow Estimation by Growing Correspondence Seeds”, *in IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[15] C. Vogel, K. Schindler and S. Roth, ”3D Scene Flow Estimation with a Rigid Motion Prior”, *in IEEE International Conference on Computer Vision*, 2011.

[16] B. Ranft and O. Denninger, ”Run-time Adaptation to Heterogeneous Processing Units for Real-time Stereo Vision”, *in IEEE International Conference on Embedded Software and Systems*, 2012.

[17] A. Geiger, P. Lenz and R. Urtasun, ”Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite”, *in IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[18] H. Sahlbach, R. Ernst, S. Wonneberger and T. Graf, ”Exploration of FPGA-based Dense Block Matching for Motion Estimation and Stereo Vision on a Single Chip”, *in IEEE Intelligent Vehicles Symposium*, 2011.

[19] B. Ranft and O. Denninger, ”A Framework for On-line Optimization of Performance and Energy Efficiency on Heterogeneous Systems”, *in IEEE International Parallel & Distributed Processing Symposium*, 2014.