The Stixel World - A Compact Medium Level Representation of the 3D-World

Hernán Badino[1], Uwe Franke[2], and David Pfeiffer[2]
hbadino@cs.cmu.edu, {uwe.franke,david.pfeiffer}@daimler.com

[1] Goethe University Frankfurt**
[2] Daimler AG

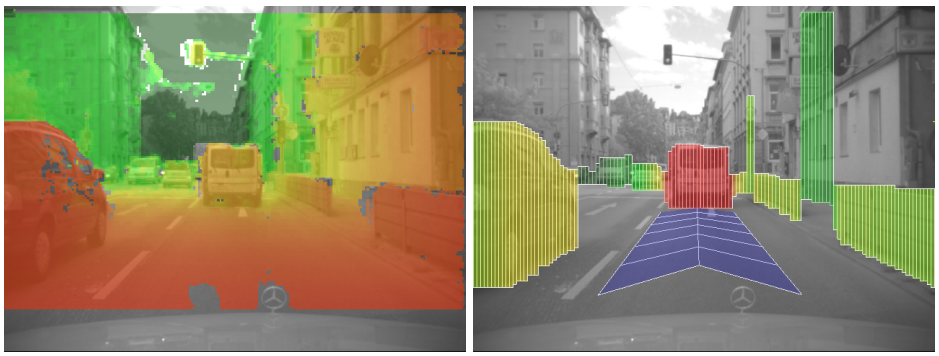
Abstract. Ambitious driver assistance for complex urban scenarios demands a complete awareness of the situation, including all moving and stationary objects that limit the free space. Recent progress in real-time *dense stereo vision* provides precise depth information for nearly every pixel of an image. This rises new questions: How can one efficiently analyze half a million disparity values of next generation imagers? And how can one find all relevant obstacles in this huge amount of data in real-time? In this paper we build a medium-level representation named "stixel-world". It takes into account that the free space in front of vehicles is limited by objects with almost vertical surfaces. These surfaces are approximated by adjacent rectangular sticks of a certain width and height. The stixel-world turns out to be a compact but flexible representation of the three-dimensional traffic situation that can be used as the common basis for the scene understanding tasks of driver assistance and autonomous systems.

1 Introduction

Stereo vision will play an essential role for scene understanding in cars of the near future. Recently, the dense stereo algorithm "Semi-Global Matching" (SGM) has been proposed [1], which offers accurate object boundaries and smooth surfaces. According to the Middlebury data base, three out of the ten most powerful stereo algorithms are currently SGM variants. Due to the computational burden, in particular the required memory bandwidth, the original SGM algorithm is still too complex for a general purpose CPU. Fortunately, we were able to implement an SGM variant on an FPGA (Field Programmable Gate Array).

The task at hand is to extract and track every object of interest captured within the stereo stream. The research of the last decades was focused on the detection of cars and pedestrians from mobile platforms. It is common to recognize different object classes independently. Therefore the image is evaluated repetitively. This common approach results in complex software structures, which remain incomplete in detection, since only objects of interest are observed. Aiming at a generic vision system architecture for driver assistance, we suggest the

** Hernán Badino is now with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA.



(a) Dense disparity image (SGM result) (b) Stixel representation

Fig. 1. (a) Dense stereo results overlaid on the image of an urban traffic situation. The colors encode the distance, red means close, green represents far. Note that SGM delivers measurements even for most pixels on the road. (b) Stixel representation for this situation. The free space (not explicitly shown) in front of the car is limited by the stixels. The colors encode the lateral distance to the expected driving corridor shown in blue.

use of a medium level representation that bridges the gap between the pixel and the object level.

To serve the multifaceted requirements of automotive environment perception and modeling, such a representation should be:

- *compact:* offering a significant reduction of the data volume,
- *complete:* information of interest is preserved,
- *stable:* small changes of the underlying data must not cause rapid changes within the representation,
- *robust:* outliers must have minimal or no impact on the resulting representation.

We propose to represent the 3D-situation by a set of rectangular sticks named "stixels" as shown in Fig. 1(b). Each stixel is defined by its 3D position relative to the camera and stands vertically on the ground, having a certain height. Each stixel limits the free space and approximates the object boundaries. If for example, the width of the stixels is set to 5 pixels, a scene from a VGA image can be represented by 640/5=128 stixels only.

Observe, that a similar stick scheme was already formulated in [2] to represent and render 3D volumetric data at high compression rates. Although our stixels are different to those presented in [2], the properties of compression, compactness and exploitation of the spatial coherence are common in both representations.

The literature provides several object descriptors like particles [3], quadtrees, octtrees and quadrics [4] [5], patchlets [6] or surfels [7]. Even though these structures partly suffice our designated requirements, we refrain from their usage for our matter since they do not achieve the degree of compactness we strive for.

Section 2 describes the steps required to build the stixel-world from raw stereo data. Section 3 presents results and properties of the proposed representation. Future work is discussed in Section 4 and Section 5 concludes the paper.

2 Building the Stixel-World

Traffic scenes typically consist of a relatively planar free space limited by 3D obstacles that have a nearly vertical pose. Fig. 1 displays a typical disparity input and the resulting stixel-world. The different steps applied to construct this representation are depicted in Fig. 2 and Fig. 3. An occupancy grid is computed from the stereo data (see Fig. 2(a)) and used for an initial free space computation. We formulate the problem in such a way that we are able to use dynamic programming which yields a global optimum. The result of this step is shown in Fig. 2(c) and 3(a). By definition, the free space ends at the base-point of vertical obstacles. Stereo disparities vote for their membership to the vertical obstacle generating a membership cost image (Fig. 3(c)).

A second dynamic programming pass optimally estimates the height of the obstacles. An appropriate formulation of this problem allows us to reuse the same dynamic programming algorithm for this task, which was applied for the free space computation. The result of the height estimation is depicted in Fig. 3(d). Finally, a robust averaging of the disparities of each stixel yields a precise model of the scene.

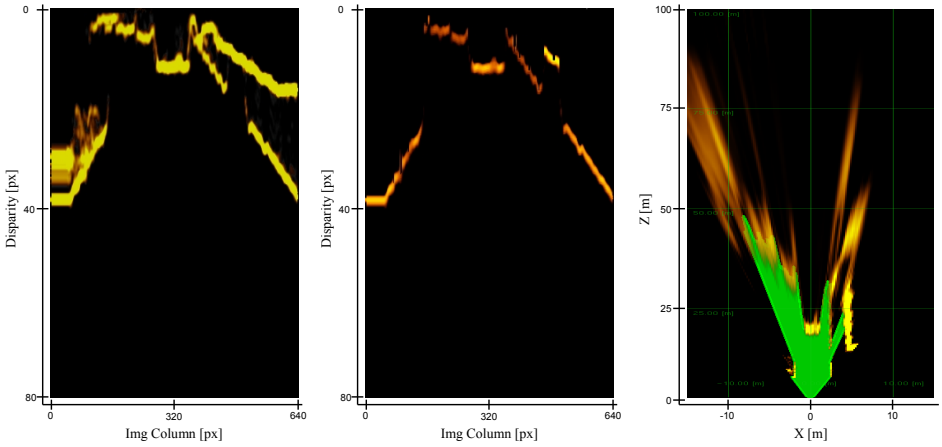
2.1 Dense Stereo

Most real-time stereo algorithms based on local optimization techniques deliver sparse disparity data. Hirschmüller [1] proposed a dense stereo scheme named "Semi-Global Matching" that runs within a few seconds on a PC. For road scenes, the "Gravitational Constraint" has been introduced in [8] which improves the results by taking into account that the disparities tend to increase monotonously from top to bottom. The implementation of this stereo algorithm on a FPGA allows us to run this method in real-time. Fig. 1(a) shows that SGM is able to model object boundaries precisely. In addition, the smoothness constraint used in the algorithm leads to smooth estimations in low contrast regions, exemplarily seen on the street and the untextured parts of the vehicles and buildings.

2.2 Occupancy Grid

The stereo disparities are used to build a stochastic occupancy grid. An occupancy grid is a two-dimensional array or grid which models occupancy evidence of the environment. Occupancy grids were first introduced in [9]. A review is given in [10].

Occupancy grids are computed in real-time using the method presented in [11] which allows to propagate the uncertainty of the stereo disparities onto



(a) Polar occ. grid. (b) Background subtraction. (c) Obtained free space.

Fig. 2. Occupancy grids: Fig. (a) shows the polar occupancy grid obtained from the disparity image shown in Fig. 1(a) (brightness encode the likelihood of occupancy). Fig. (b) shows the resulting image when background subtraction is applied to Fig. (a). The free space obtained from dynamic programming is shown in Fig. (c) in green, overlaid on a Cartesian representation of the occupancy grid.

the grid. We use a polar occupancy grid in which the image column is used to represent the angular coordinate and the stereo disparity is used to represent the range. Figure 2(a) shows an example of a the polar occupancy grid obtained from the stereo result shown in Fig. 1(a).

Only those 3D measurements lying above the road are registered as obstacles in the occupancy grid. Instead of assuming a planar road, we estimate the road pose by fitting a B-Spline surface to the 3D data as proposed in [12].

2.3 Free space computation

The task in free space analysis is to find the first visible relevant obstacle in the positive direction of depth. By observing Fig. 2(a) this means that the search must start from the bottom of the image in vertical direction until an occupied cell is found. The space found in front of this cell is considered free space. Instead of using a thresholding operation for every column independently, we use dynamic programming (DP) to find the optimal path cutting the polar grid from left to right. As proposed in [11], spatial smoothness is imposed by using a cost that penalizes jumps in depth, while temporal smoothness is imposed by a cost that penalizes the deviation of the current solution from a prediction. The prediction is obtained from the segmentation result of the previous cycle.

In real world scenes, an image column may contain more than one object. In the example considered here, the guardrail at the right and the building at the background in Fig. 1, both have a corresponding occupancy likelihood in the

occupancy grid of Fig. 2(a). Nevertheless, per definition, the free space is given only up to the guardrail. Applying dynamic programming directly on the grid of Fig. 2(a) might lead to a solution where the optimal boundary is found on the background object (i.e. the building) and not on the foreground object (i.e. the guardrail).

To cope with the above problem, a background subtraction is carried out before applying DP. All occupied cells behind the first maximum which is above a given threshold are marked as free. The threshold must be selected so that it is quite larger than the occupancy grid noise expected in the grid. An example of the resulting background subtraction is shown in Fig. 2(b).

The output of the DP is a set of vector coordinates $(u, \hat{d}_u)$, where u is a column of the image and $\hat{d}_u$ the disparity corresponding to the distance up to which free space is available. For every pair $(u, \hat{d}_u)$ a corresponding triangulated pair (x_u, z_u) is computed, which defines the 2D world point corresponding to $(u, \hat{d}_u)$. The sorted collection of points (x_u, z_u) plus the origin $(0, 0)$ form a polygon which defines the free space area from the camera point of view (see Fig. 2(c)). Fig. 3(a) shows the free space overlaid on the left image when dynamic programming is applied on Fig. 2(b).

Observe that each free space point of the polygon in Fig. 3(a) indicates not only the interruption the free space but also the base-point of a potential obstacle located at that position (a similar idea was successfully applied in [13]). The next section describes how to apply a second pass of dynamic programming in order to obtain the upper boundary of the obstacle.

2.4 Height Segmentation

The height of the obstacles is obtained by finding the optimal segmentation between foreground and background disparities. This is achieved by first computing a cost image and then applying dynamic programming to find the upper boundary of the objects.

Given the set of points $(u, \hat{d}_u)$ and their corresponding triangulated coordinate vectors (x_u, z_u) obtained from the free space analysis, the task is to find the optimal row position v_t where the upper boundary of the object at (x_u, z_u) is located.

In our approach every disparity $d(u, v)$ (i.e. disparity on column u and row v) of the disparity image votes for its membership to the foreground object. In the simplest case a disparity votes positively for its membership as belonging to the foreground object if it does not deviate more than a maximal distance from the expected disparity of the object. The disparity votes negatively otherwise. The Boolean assignments make the threshold for the distance very sensitive: if it is too large, all disparities vote for the foreground membership, if it is too small, all points vote for the background. A better alternative is to approximate the Boolean membership in a continuous variation with an exponential function of the form

$$M_{u,v}(d) = 2^{\left(1-\left(\frac{d-\hat{d}_u}{\Delta D_u}\right)^2\right)} - 1 \tag{1}$$

where ΔD_u is a computed parameter and $\hat{d}_u$ is the disparity obtained from the free space vector (Sec. 2.3), i.e. the initially expected disparity of the foreground object in the column u. The variable ΔD_u is derived for every column independently as

$$\Delta D_u = \hat{d}_u - f_d(z_u + \Delta Z_u), \quad \text{where} \quad f_d(z) = \frac{b \cdot f_x}{z} \tag{2}$$

and $f_d(z)$ is the disparity corresponding to depth z. b corresponds to the baseline, f_x is the focal length and ΔZ_u is a parameter. This strategy has the objective to define the membership as a function in meters instead of pixels to correct for perspective effects. For the results shown in this paper we use $\Delta Z_u = 2\,\text{m}$. Fig. 3(b) shows an example of the membership values. Our experiments show that the explicit choice of the functional is not crucial as long as it is continuous.

From the membership values the cost image is computed:

$$C(u,v) = \sum_{i=0}^{i=v-1} M_{u,v}(d(u,i)) - \sum_{i=v}^{i=v_f} M_{u,v}(d(u,i)) \tag{3}$$

where v_f is the row position such that the triangulated 3D position of disparity $\hat{d}_u$ on image position (u, v_f) lies on the road, i.e. is the row corresponding to the base-point of the object. Fig. 3(c) shows an exemplary cost image.

For the computation of the optimal path, a graph $G_{hs}(V_{hs}, E_{hs})$ is generated. V_{hs} is the set of vertices and contains one vertex for every pixel in the image. E_{hs} is the set of edges which connect every vertex of one column with every vertex of the following column.

The cost minimized by dynamic programming is composed of a data and a smoothness term, i.e.;

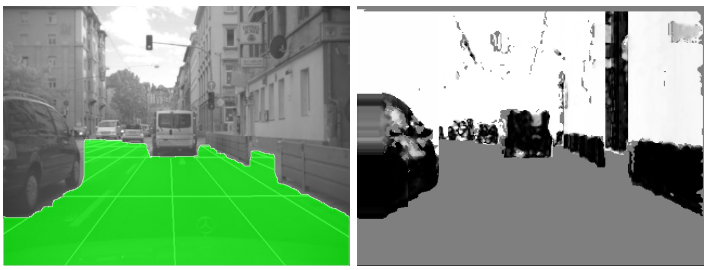
$$c_{u,v_0,v_1} = C(u,v_0) + S(u,v_0,v_1) \tag{4}$$

is the cost of the edge connecting the vertices V_{u,v_0} and V_{u+1,v_1} where $C(u,v)$ is the data term as defined in Eq. 3. $S(u,v_0,v_1)$ applies smoothness and penalizes jumps in the vertical direction and is defined as:

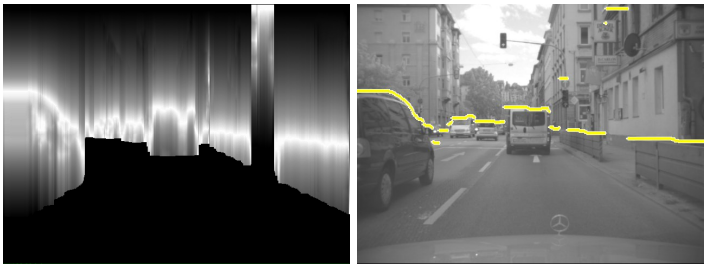
$$S(u,v_0,v_1) = C_s|v_0 - v_1| \cdot \max\left(0, 1 - \frac{|z_u - z_{u+1}|}{N_Z}\right) \tag{5}$$

where C_s is the cost of a jump. The cost of a jump is proportional to the difference between the rows v_0 and v_1. The last term has the effect of relaxing the smoothness constraint at depth discontinuities. The spatial smoothness cost of a jump becomes zero if the difference in depth between the columns is equal or larger than N_Z. The cost reaches its maximum C_s when the free space distance between consecutive columns is 0. For our experiments we use $N_Z = 5\,\text{m}$, $C_s = 8$.

An exemplary result of the height segmentation for the free space computed in Fig. 3(a) is shown in Fig. 3(d).



(a) Free space (b) Membership values



(c) Membership cost image (d) Height segmentation

Fig. 3. Stixels computation: Fig. (a) shows the result obtained from free space computation with dynamic programming. The assigned membership values for the height segmentation are shown in Fig. (b), while the cost image is shown in Fig. (c) (the grey values are negatively scaled). Fig. (d) shows the resulting height segmentation.

2.5 Stixel Extraction

Once the free space and the height for every column has been computed, the extraction of the stixel is straightforward. If the predefined width of the stixel is more than one column, the heights obtained in the previous step are fused resulting in the height of the stixel. The parameters base and top point v_B and v_T as well as the width of the stixel span a frame where the stixel is located.

Due to discretization effects of the free space computation, which are caused by the finite resolution of the occupancy grid, the free space vector is condemned to a limited accuracy in depth. Further spatial integration over disparities within this frame grant an additional gain in depth accuracy. The disparities found within the stixel area are registered in a histogram while regarding the depth uncertainty known from SGM. A parabolic fit around the maximum delivers the new depth information. This approach offers outlier rejection and noise suppression, which is illustrated by Fig. 4, where the SGM stereo data of the rear of a truck are displayed. Assuming a disparity noise of $0.2\,px$, a stereo baseline of 0.35 m and a focal length of $830\,px$, as in our experiments, the expected standard deviation for the truck at 28 meters is approx. 0.54 m. Since an average stixel covers hundreds of disparity values, the integration significantly improves the

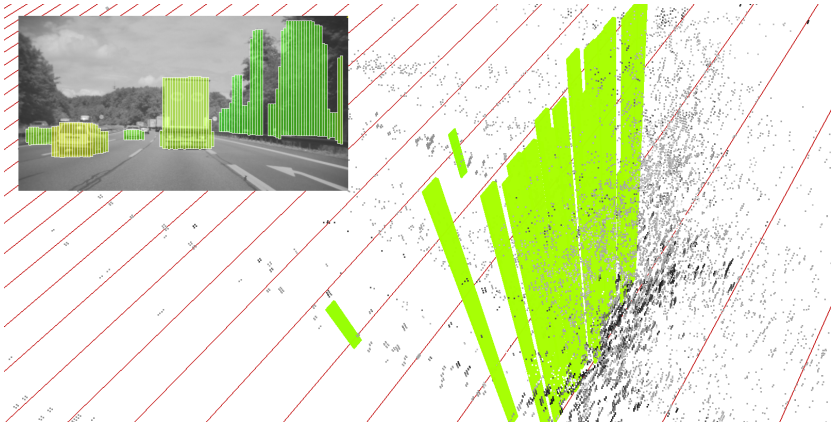


Fig. 4. 3D visualization of the raw stereo data showing a truck driving 28 meters ahead. Each red line represents 1 meter in depth. One can clearly observe the high scattering of the raw stereo data while the stixels remain as a compound and approximate the planar rear of the truck.

depth of the stixel. As expected, the uncertainty falls below 0.1m for each stixel.

3 Experimental Results

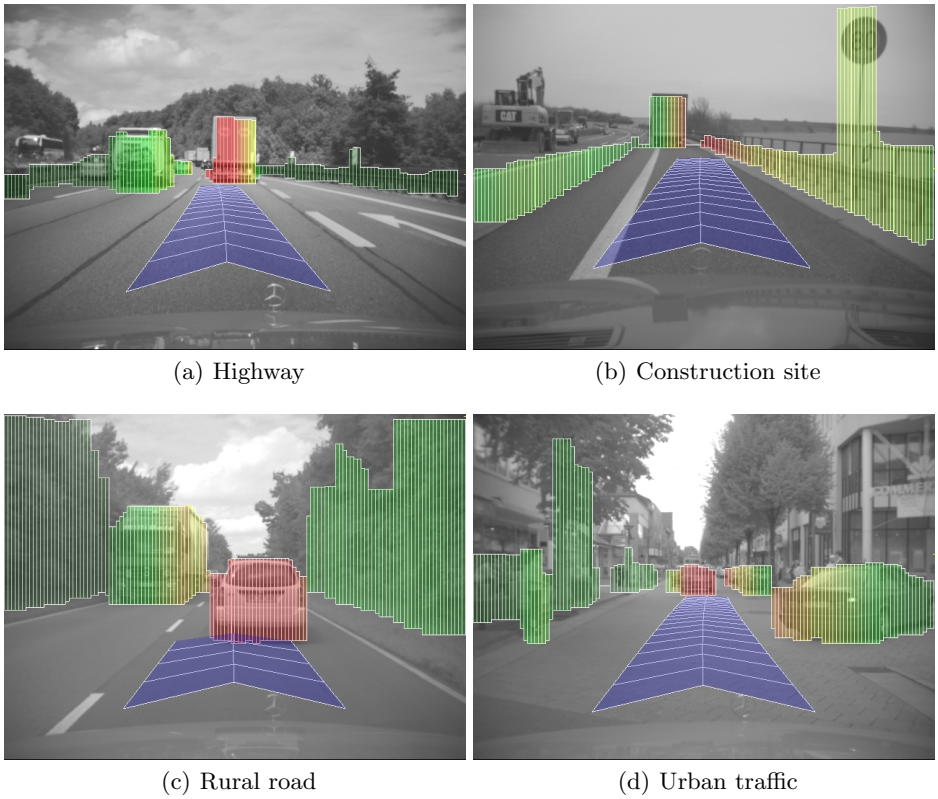
Figure 5 displays the results of the described algorithm when applied to images taken from different road scenarios such as highways, construction sites, rural roads and urban environments. The stereo baseline is 0.35 m, the focal length 830 pixels and the images have a VGA (640×480 pixels) resolution.

The color of the stixels encodes the lateral distance to the expected driving corridor. It's highly visible that even filigree structures like beacons or reflector posts are being captured in their position and extension. For clarity reasons we do not explicitly show the obtained free space.

The complete computation of stixels on a Intel Quad Core 3.00 GHz processor takes less than 25 milliseconds. The examples shown in this paper must be taken as representative results of the proposed approach. In fact, the method has successfully passed days of real-time testing in our demonstrator vehicle in urban, highway and rural environments.

4 Future work

In the future we intend to apply a tracking for stixels based upon the principles of 6D-Vision [14], where 3D points are tracked over time and integrated with Kalman filters. The integration of stixels over time will lead to further improvement of the position and height. At the same time it will be possible to estimate the velocity and acceleration, which will ease subsequent object clustering steps.



(a) Highway (b) Construction site

(c) Rural road (d) Urban traffic

Fig. 5. Evaluation of stixels in different real world road scenarios showing a highway, a construction site, a rural road and an urban environment. The color encodes the lateral distance to the driving corridor. Base points (i.e. distance) and height estimates are in very good accordance to the expectation.

Almost all objects of interest within the dynamic vehicle environment touch the ground. Nevertheless, hovering or flying objects such as traffic signs, traffic lights and side mirrors (an example is given in Fig. 5(b) at the traffic sign) violate this constraint. Our efforts in the future work also includes to provide a dynamic height of the base-point.

5 Conclusion

A new primitive called stixel was proposed for modeling 3D scenes. The resulting *stixel-world* turns out to be a robust and very compact representation (not only) of the traffic environment, including the free space as well as static and moving objects.

Stochastic occupancy grids are computed from dense stereo information. Free space is computed from a polar representation of the occupancy grid in order

to obtain the base-point of the obstacles. The height of the stixels is obtained by segmenting the disparity image in foreground and background disparities applying the same dynamic programming scheme as used for the free space computation. Given height and base point the depth of the stixel is obtained with high accuracy.

The proposed stixel scheme serves as a well formulated medium-level representation for traffic scenes. Obviously, the presented approach is also promising for other applications that obey the same assumptions of the underlying scene structure.

Acknowledgment

The authors would like to thank Jan Siegemund for his contribution to the literature review and Stefan Gehrig and Andreas Wedel for fruitful discussions.

References

1. Hirschmüller, H.: Accurate and efficient stereo processing by semi-global matching and mutual information. In: CVPR. (2005)
2. Montani, C., Scopigno, R.: Rendering volumetric data using the sticks representation scheme. In: Workshop on Volume Visualization, San Diego, California (1990)
3. Fua, P.: Reconstructing complex surfaces from multiple stereo views. In: ICCV. (June 1996)
4. Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., Stuetzle, W.: Surface reconstruction from unorganized points. In: Conference on Computer Graphics and Interactive Techniques. (1992) 71–78
5. Ohtake, Y., Belyaev, A., Alexa, M., Turk, G., Seidel, H.P.: Multi-level partition of unity implicits. ACM SIGGRAPH 2003 **22**(3) (2003) 463–470
6. Murray, D., Little, J.J.: Segmenting correlation stereo range images using surface elements. In: 3D Data Processing, Visualization and Transmission. (September 2004) 656–663
7. Pfister, H., Zwicker, M., van Baar, J., Gross, M.: Surfels: Surface elements as rendering primitives. In: ACM SIGGRAPH. (2000)
8. Gehrig, S., Franke, U.: Improving sub-pixel accuracy for long range stereo. In: VRML Workshop, ICCV. (2007)
9. Elfes, A.: Sonar-based real-world mapping and navigation. Journal of Robotics and Automation **3**(3) (June 1987) 249–265
10. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. Intelligent Robotics and Autonomous Agents. The MIT Press (2005)
11. Badino, H., Franke, U., Mester, R.: Free space computation using stochastic occupancy grids and dynamic programming. In: Workshop on Dynamical Vision, ICCV, Rio de Janeiro, Brazil (October 2007)
12. Wedel, A., Franke, U., Badino, H., Cremers, D.: B-spline modeling of road surfaces for freespace estimation. In: Intelligent Vehicle Symposium. (2008)
13. Kubota, S., Nakano, T., Okamoto, Y.: A global optimization algorithm for real-time on-board stereo obstacle detection systems. In: Intelligent Vehicle Symposium. (2007)
14. Franke, U., Rabe, C., Badino, H., Gehrig, S.: 6d-vision: Fusion of stereo and motion for robust environment perception. In: DAGM. (2005)