

Free Space Computation Using Stochastic Occupancy Grids and Dynamic Programming

Hernán Badino¹, Uwe Franke², Rudolf Mester¹

¹ Johann Wolfgang Goethe University, Frankfurt am Main

² DaimlerChrysler AG, Stuttgart

Abstract. The computation of free space available in an environment is an essential task for many intelligent automotive and robotic applications. This paper proposes a new approach, which builds a stochastic occupancy grid to address the free space problem as a dynamic programming task. Stereo measurements are integrated over time reducing disparity uncertainty. These integrated measurements are entered into an occupancy grid, taking into account the noise properties of the measurements. In order to cope with real-time requirements of the application, three occupancy grid types are proposed. Their applicabilities and implementations are also discussed. Experimental results with real stereo sequences show the robustness and accuracy of the method. The current implementation of the method runs on off-the-shelf hardware at 20 Hz.

1 Introduction

The computation of free space available in the environment is an essential task for many intelligent automotive and robotic applications. The free space is the world regions where navigation without collision is guaranteed. Navigable space might become extremely important in automotive applications if an escape route in a critical situation is required. In robotics, free space is required when planning the path between two points. This paper proposes a method for the computation of free space in complex traffic scenarios.

Literature Review. Occupancy grids were first introduced in [2]. Occupancy grids were proposed as a model to handle a number of problems in the mobile robot domain. This includes: range-based mapping, multiple sensor integration, path planning, and navigation. Although the grids were used initially to model occupancy probabilities, the encoding of multiple properties in the cell state was proposed early in [2]. In [5] and [9] three-dimensional grids were used to model properties like color, texture, and occupancy. Occupancy or evidence grids continue to be used everywhere in the literature for spatial modelling of environments. An exhaustive review can be found in [13] and [14].

An occupancy grid is built in [12] based on the input given by a stereo camera. 3D points are tracked over time and integrated with Kalman filters. The estimated covariance matrix of the 3D points is used to update the occupancy

grid. However, the error of the triangulated 3D points is modelled as a Gaussian distribution. This leads to biased estimates of 3D position [11], and therefore to the wrong estimation of free space. In [10] the free space is computed using occupancy grids and stereo vision. The method uses filters to improve the quality of the disparity image and apply a threshold to decide the occupancy or non-occupancy of the cells. Free space computation on occupancy grids were also addressed by [8] using laser range finders and omnidirectional stereo. In [1] free space is computed independently of evidence grids by applying inverse perspective mapping. [4] indirectly obtains free space by computing the road-obstacle boundaries by dynamic programming on a simplified disparity space image.

This paper presents a novel method for the computation of free space based on dynamic programming on a polar occupancy grid. The Cartesian representation of an occupancy grid is redefined in this paper and two additional types of grids are proposed.

2 Overview of the Algorithm

Figure 1 shows a block diagram of the proposed algorithm. The algorithm starts by computing stereo with input from a pair of rectified stereo images. The stereo algorithm corresponding to the *Stereo Computation* block is not constrained to a specific implementation. The only requirement is computation of enough disparity measurements to capture all relevant objects in a scene. The stereo algorithm generates a disparity and a variance image. The variance image contains the estimated variance of each measured disparity.

Disparity and variance images are used to refine an integrated disparity image by means of Kalman filters. An iconic representation (section 4) is used, where the state vector of the filter collects all disparity measurements. The ego-motion of the camera must be available in order to predict the disparity and variance images. The prediction is corrected with a new stereo measurement to obtain

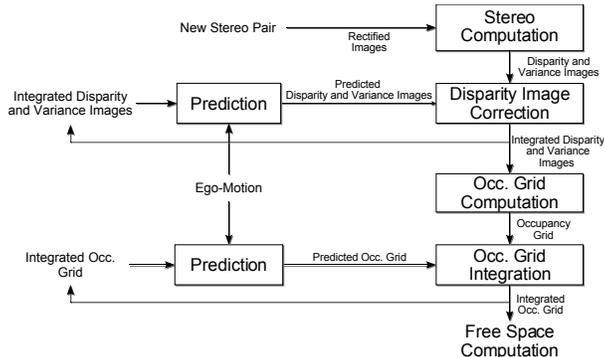


Fig. 1. Block diagram of the algorithm.

refined disparity and variance images. The next step is to compute the occupancy grid. Section 3 addresses this topic.

The occupancy grid is also integrated over time to reduce the effect of outliers. The *Occupancy Grid Integration* block performs a low-pass filter with a predicted occupancy grid based on ego-motion of the camera. Finally, free space is computed using dynamic programming on a polar grid (topic of section 5).

The next section deals with occupancy grids and how to generate them from stereo measurements.

3 Occupancy Grids

Definition. An occupancy grid M is a two-dimensional array or grid which models occupancy evidence of the environment. The 3D world is orthographically projected on a plane \mathcal{P} parallel to the road (assuming a structured environment where the floor surface is planar). The plane is discretized into cells (i, j) . Every cell corresponds to some tetragonal area A_{ij} of the plane \mathcal{P} . The intersection area of any two cells is always empty, i.e. $A_{ij} \cap A_{lm} = \emptyset, (i, j) \neq (l, m)$. The areas A_{ij} are not necessarily rectangular and may not be equal. The subindex ij specifies a lateral component (i) and a depth component (j). Every cell of the grid maintains an *occupancy likelihood* $D(i, j)$ of the represented world area. Figure 2 shows some examples of occupancy grids. The following sections present three types of occupancy grids based on the this definition.

Mathematical Preliminaries. A measurement is defined as the vector $\mathbf{m}_k = (u, v, d)^T$, where (u, v) is a left image coordinate, and d its corresponding disparity computed by stereo. The measurement \mathbf{m}_k is the projection of some world feature located at point $\mathbf{p}_k = (x, y, z)^T$ onto the camera image, such that $\mathbf{m}_k = \mathbf{P}(\mathbf{p}_k)$. The projection equation is;

$$\mathbf{m}_k = \mathbf{P}(\mathbf{p}_k) = \frac{1}{z} \begin{pmatrix} f_u x \\ f_v y \\ f_u B \end{pmatrix} + \begin{pmatrix} u_0 \\ v_0 \\ 0 \end{pmatrix}, \quad (1)$$

where f_u and f_v are the focal lengths measured in pixel width and height respectively. B is the baseline of the stereo system, and (u_0, v_0) is the principal point of the image. The inverse of the projection equation is the triangulation equation which is defined as;

$$\mathbf{p}_k = \mathbf{P}^{-1}(\mathbf{m}_k) = \frac{B}{d} \begin{pmatrix} (u - u_0) \\ (v - v_0)s_{vu} \\ f_u \end{pmatrix}, \quad (2)$$

where $s_{vu} = f_u/f_v$.

The vector \mathbf{m}_k is a noisy measurement of a real but unknown vector $\bar{\mathbf{m}}_k$, such that $\bar{\boldsymbol{\xi}}_k = \mathbf{m}_k - \bar{\mathbf{m}}_k$ is the real unknown error vector. $\bar{\boldsymbol{\xi}}_k$ is assumed to be an occurrence of a zero mean random process with probability density function

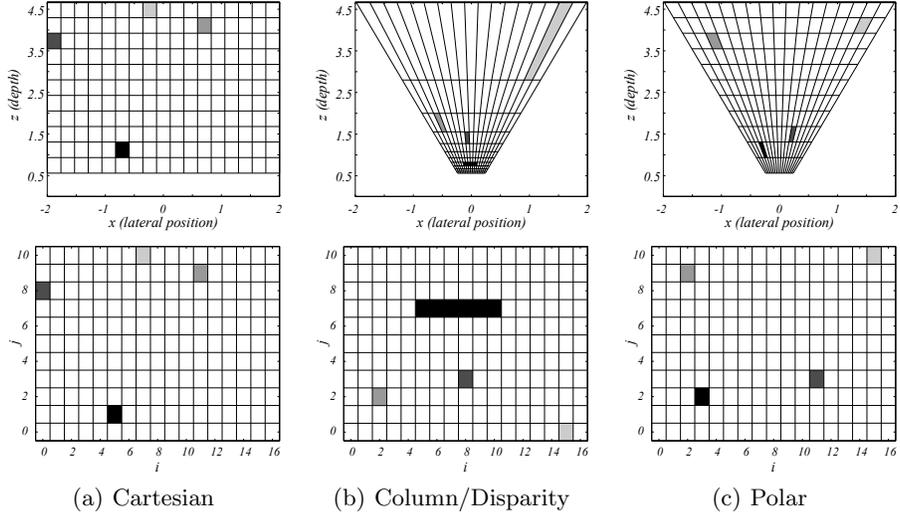


Fig. 2. Examples of occupancy grids. The figures on the top show the plane \mathcal{P} with the discretized areas A_{ij} . The figures on the bottom show the corresponding occupancy grids. Some cells and their corresponding areas have been marked in order to show the world areas represented in each case.

(p.d.f.) $\mathbf{G}_{\bar{\mathbf{m}}_k}$, i.e. $\mathbf{G}_{\bar{\mathbf{m}}_k}(\boldsymbol{\xi}_k)$ models the likelihood of obtaining an error $\boldsymbol{\xi}_k$ given the fact that the real state is $\bar{\mathbf{m}}_k$. Without loss of generality it is assumed a multivariate Gaussian p.d.f.;

$$\mathbf{G}_{\bar{\mathbf{m}}_k}(\boldsymbol{\xi}_k) = \frac{1}{(2\pi)^{3/2} |\bar{\mathbf{\Gamma}}_k|} \exp\left(-\frac{1}{2} \boldsymbol{\xi}_k^T \bar{\mathbf{\Gamma}}_k^{-1} \boldsymbol{\xi}_k\right), \quad (3)$$

where $\bar{\mathbf{\Gamma}}_k$ is the real measurement covariance matrix.

The objective is to find the function $L_{ij}(\mathbf{m}_k)$ which defines the occupancy likelihood for cell (i, j) given measurement \mathbf{m}_k . The obtained likelihood is added to the current cell value of the occupancy grid $D(i, j)$, such that for m measurements the occupancy likelihood for a given cell (i, j) is:

$$D(i, j) = \sum_{k=1}^m L_{ij}(\mathbf{m}_k). \quad (4)$$

In the following sections three types of occupancy grids are defined, and the corresponding L_{ij} is found.

Cartesian Occupancy Grid. The world is represented by a Cartesian grid, i.e. a portion of the world is projected to a plane, and mapped linearly to a grid of fixed dimensions (see Figure 2(a)). Assuming that cell (i, j) of the Cartesian

grid is centered at world coordinate (x_{ij}, z_{ij}) , and that $\bar{\mathbf{m}}_k$ can be approximated by \mathbf{m}_k the likelihood function for cell (i, j) is;

$$L_{ij}(\mathbf{m}_k) = \mathbf{G}_{\mathbf{m}_k}(\mathbf{P}(\mathbf{p}_{ij}) - \mathbf{m}_k), \quad (5)$$

where $\mathbf{p}_{ij} = (x_{ij}, y, z_{ij})^T$ and y is the triangulated height of the measurement obtained with Equation 2. Equation 5 defines likelihood of the cell to be; Gauss factor dependent on deviation between measurement and the projected cell position. Thus, the maximum likelihood factor is given to the cell which contains the triangulated measurement (see Figure 3(a)).

In the actual implementation, the registration of measurement \mathbf{m}_k by updating *every* cell of the grid is time consuming, and prohibitive for real-time application. A more appropriate implementation updates only the cells which are affected significantly by the current measurement. For example, when assuming Gaussian p.d.f., only cells for which the Mahalanobis distance between its projection, and the measurement is less than 3 are considered. The amount of measurement data not registered is less than 0.3%.

Column/Disparity Map. Contribution of the v component of measurement vector \mathbf{m}_k in Equation 1 can be ignored since it contributes only to the height component of the 3D point \mathbf{p}_k . Since the height component is lost by the projection onto the grid, the vector (u, d) suffices to register the measurement. The cells of the column/disparity grid correspond to discretized values of the u and d image coordinates (see Figure 2(b)). Assuming that the cell (i, j) corresponds to the coordinate (u_{ij}, d_{ij}) , The likelihood function for the cell (i, j) is:

$$L_{ij}(\mathbf{m}_k) = \mathbf{G}_{\mathbf{m}_k}((u_{ij} - u, 0, d_{ij} - d)^T). \quad (6)$$

Figure 3(b) shows an example of the L_{ij} function.

Polar Occupancy Grid. In stereo triangulation, depth varies inversely proportional with disparity. Registering stereo measurements in a column/disparity grid implies a decreasing resolution to distant points (compare Figures 2(b) and 2(c)). This problem is overcome by defining a polar grid. In a polar occupancy grid, cells represent discretized values of coordinates (u, z) where u corresponds to the column of the image, and z the depth in the world coordinate system. Let us assume that cell (i, j) corresponds to coordinate (u_{ij}, z_{ij}) . The likelihood function for cell (i, j) is;

$$L_{ij}(\mathbf{m}_k) = \mathbf{G}_{\mathbf{m}_k}((u_{ij} - u, 0, d'_{ij} - d)^T), \quad (7)$$

where $d'_{ij} = f_u B / z_{ij}$, i.e. disparity corresponding to the cell depth.

Discussion The usual representation is a Cartesian grid [14]. It offers an intuitive way to model the environment since there is linear mapping between world

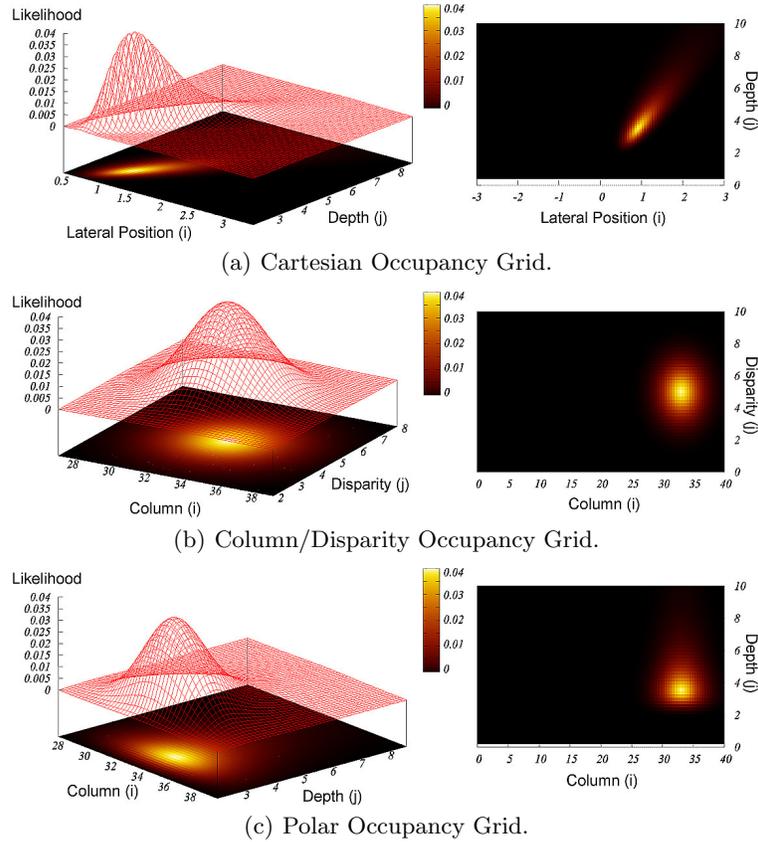


Fig. 3. Probability density functions for the same measurement in each occupancy grid representation. The figures show the likelihood function L_{ij} .

and occupancy grid coordinates. Nevertheless, a Cartesian representation is not always the best choice. The main drawback of a Cartesian grid is its computation time. Every measurement has a different L_{ij} and affects a different number of cells. Distant points require much more registration time than close points because of the greater number of cells affected by the measurement.

A much faster implementation is obtained with a column/disparity grid. If the measurement covariance matrix $\bar{\mathbf{T}}_k$ is constant, a single look-up table stores the coefficients required to register all measurements. If $\bar{\mathbf{T}}_k$ is not constant, the computation is still faster than in the Cartesian case. The L_{ij} function is symmetric with respect to both grid axes (see figure 3(b)). This allows the computation of coefficients for only one quarter of the cells; the remaining three quarters are just mirrored transformations of the first quarter. Furthermore, the number of cells affected by a measurement does not depend on the measurement itself, but only on the covariance matrix $\bar{\mathbf{T}}_k$.

The depth resolution of a column/disparity grid decreases quadratically with distance. This might be convenient for applications requiring a higher accuracy for closer objects, but is a problem if the goal is detection of distant objects. A polar occupancy grid provides a constant depth resolution at the expense of some computation time. The form of the likelihood function L_{ij} is asymmetric in the depth direction (because of non-linearity of the triangulation equation) but symmetric in the horizontal direction (see Figure 3(c)). This means that only half of the cells affected by a measurement require computation. If the covariance matrix is constant, maintaining a look-up table for every row in the grid speeds up the algorithm.

It is convenient be able to transform one representation into another representation. In section 5 this will be required for the computation of free space using dynamic programming. Given any two occupancy grids of different types D^a and D^b , where a and b , where chosen just for convenience, and indicate any two different types of occupancy grids, the transformation is;

$$D^a(i, j) = \sum_{(l, m) \in \Omega} D^b(l, m), \quad (8)$$

such that;

$$\Omega = (l, m) \mid (l, m) \xrightarrow{T_b^a} (i, j), \quad (9)$$

and where the mapping T_b^a defines the coordinate transformation from an occupancy grid of type b to a map of type a . Ω collects all the cells corresponding to the same destination cell (i, j) .

4 Iconic Representation for Stereo Integration.

The main error of triangulated stereo measurement is in the depth component (see e.g. [6]). The reduction of disparity noise helps localization of estimated 3D points, and therefore grouping of objects in a grid. Tracking of features in an image over time allows reduction of the covariance matrix $\mathbf{\Gamma}_k$ (see [3]). Nevertheless, tracking is a very expensive operation and highly restricts real-time capability of a system with an increasing number of features. Instead a more direct method that does not require tracking, but depends on the ego-motion information of the camera, is used. Iconic representation introduced in [7] is used. In the iconic representation only disparity components of the measurement vector \mathbf{m}_l are filtered by means of a Kalman filter. The first two components of the measurement \mathbf{m}_l are discretized into integer values so that disparity measurements always correspond to integer image positions.

Stereo integration requires three main steps as figure 1 shows:

- Stereo Prediction: the current integrated disparity and variance images are predicted. This is equivalent to computing expected optical flow and disparity based on ego-motion [7]. Our prediction of the variance image includes the addition of a driving noise parameter that models the uncertainties of the system, such as ego-motion inaccuracy.

- Stereo computation: disparity and variance images are computed based on the current left and right images.
- Disparity Image Correction: both, predicted and measured disparity images are fused together obtaining the new integrated disparity and variance images. Current predicted disparities with no corresponding measurement (i.e. no disparity computed the current image position) are not immediately deleted, in the expectation that a measurement for this estimate will be computed in the next frame. Disparity measurements with no corresponding estimate are considered new, and are added into the current integrated disparity image. Every remaining measured disparity has a corresponding estimate. If the measurement does not lie within a $3I$ distance from the current estimate, the measurement is added as new replacing the estimate. Otherwise the estimate is updated with standard Kalman filter correction equations [7].

An example of the improvement achieved with the iconic representation is shown in figure 4. The occupancy grid shown in Figure 4(b) was obtained with standard output from the stereo algorithm while in Figure 4(e) the occupancy grid was computed with an integrated disparity image. The improvement can be seen by the reduction in *tails* of registered 3D points. Figure 4(f) shows the comparison when using and not using the disparity image integration. The region corresponds to the pedestrian. The integrated disparity image shows an improvement of several orders of magnitude.

5 Freespace Computation by Dynamic Programming

Cartesian space is not a suitable space to compute the free space because the search must be done in the direction of rays leaving the camera. Furthermore, the set of rays must span the whole grid. A more appropriate space is the polar space. In polar coordinates every grid column is, by definition, already in the direction of a ray. Therefore, searching for obstacles in the ray direction is straightforward. The first step is to transform the Cartesian grid to a polar grid as addressed in section 3.

In polar representation, the task is now to find the first occupied cell. The first visible obstacle in the positive direction of depth will be found. All the space found before the cell is considered free space. Figure 5(a) shows the procedure so far. By observing Figure 5(a) carefully, it can be seen that the solution forms a path from left to right segmenting transversely the polar grid in two regions. Instead of thresholding each column as usually done [10] [8], dynamic programming is used. The new method based on dynamic programming has the following properties:

- *Global optimization*: every row is not considered independently, but as part of a global optimization problem that is optimally solved.
- *Spatial and temporal smoothness of the solution*: the spatial smoothness is imposed by the use of a cost that penalizes jumps in depth while temporal

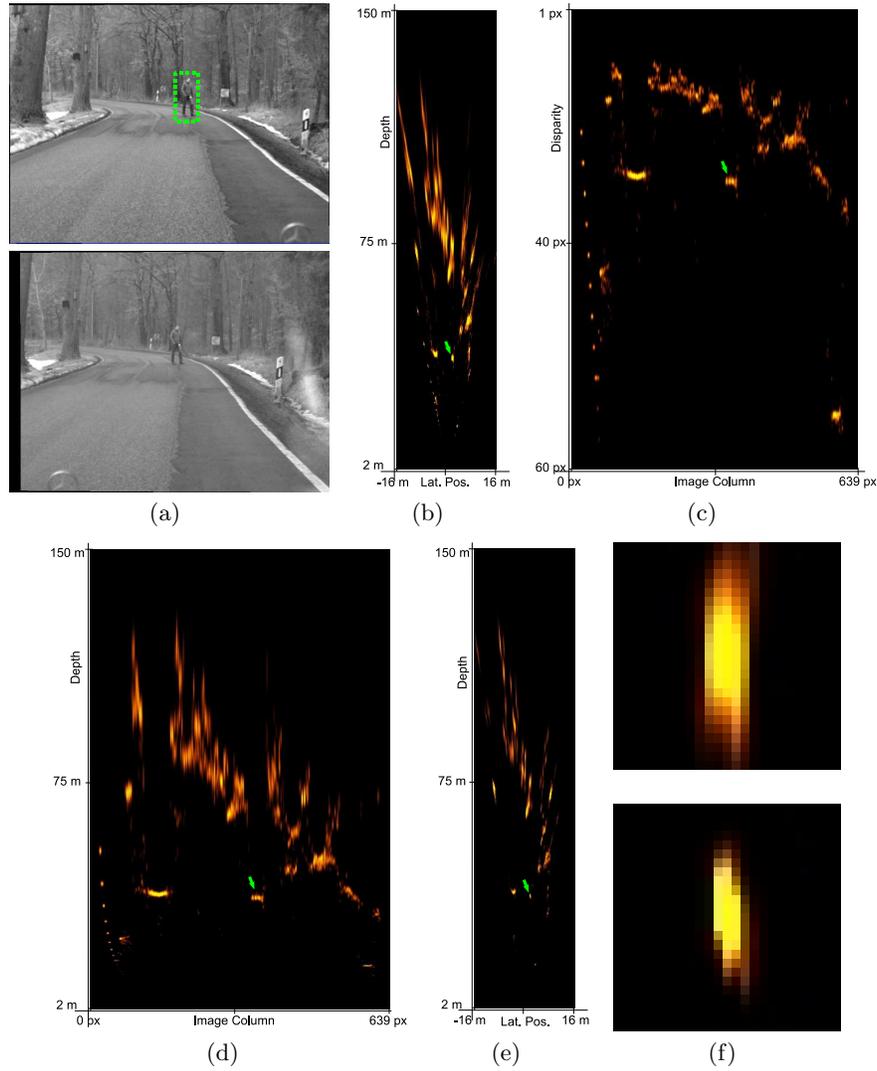


Fig. 4. Figures (b), (c) and (d) show the Cartesian, column/disparity and polar (respectively) maps computed with rectified stereo pair of Figure (a). In Cartesian space the cell size is $0.15m \times 0.15m$. The Polar grid has depth resolution of $0.15m$ and angular resolution of $1 px$. The column/disparity grid has the same angular resolution, and disparity resolution of $0.1 px$. These three occupancy grids were computed using raw stereo information. Figure (e) shows the resulting Cartesian map when integrated stereo is used as addressed in section 4. The pedestrian is shown as a reference point with a vector in all figures. Figure (f) shows amplified regions of the pedestrian likelihood for Figure (b) (top) and Figure (e) (bottom).

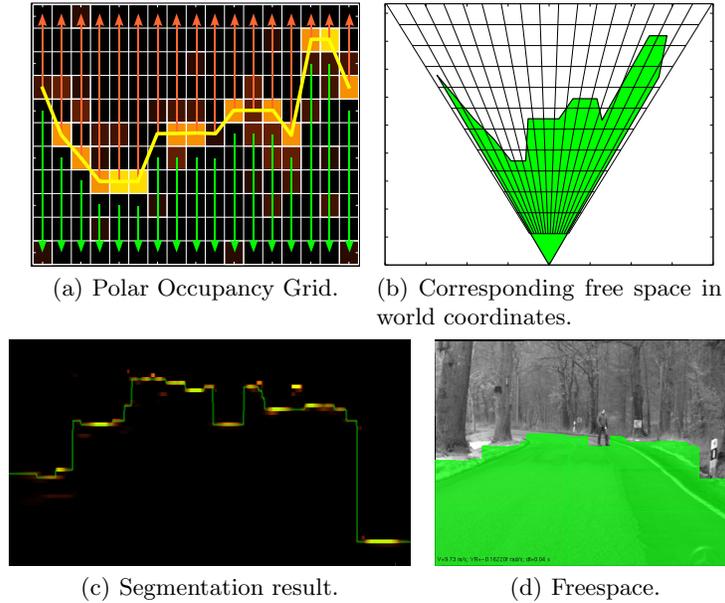


Fig. 5. Free space search in a polar representation, and corresponding free space area.

smoothness is imposed by a cost that penalizes the deviation of the current solution from a prediction.

- *Preservation of spatial and temporal discontinuities*: the saturation of the spatial and temporal costs allows the preservation of discontinuities.

Dynamic programming is applied to the grid in order to segment the image into two regions. For computation of the optimal path, a graph $G(V, E)$ is generated. V is the set of vertices, and contains one vertex for every cell in the grid. E is the set of edges which connect every vertex of one column, with every vertex of the following column. Every edge has an associated value which defines the cost of segmenting the image through the connected vertices. The objective is to find the minimal path using dynamic programming. The cost of each edge is composed of a data and a smoothness term, i.e.;

$$c_{i,j,k,l} = E_d(i,j) + E_s(i,j,k,l), \quad (10)$$

is the cost of the edge connecting the vertices V_{ij} and V_{kl} where;

$$E_d(i,j) = \frac{1}{D(i,j)}, \quad (11)$$

is the data term defined by the inverse likelihood of the cell and³ and;

$$E_s(i,j,k,l) = S(j,l) + T(i,j), \quad (12)$$

³ If $D(i,j) = 0$, then a very large value is assigned to $E_d(i,j)$

is a smoothness term containing a spatial and a temporal part. The spatial term penalizes jumps in depth and is defined as:

$$S(j, l) = \begin{cases} C_s d(j, l) & ; \text{ if } d(j, l) < T_s \\ C_s T_s & ; \text{ if } d(j, l) \geq T_s \end{cases} . \quad (13)$$

The function $d(j, l)$ returns the distance in meters between cells in rows j and l of the grid. The constant C_s is a cost parameter penalizing jumps in depth, and the threshold T_s (also measured in meters) saturates the cost function, allowing the preservation of depth discontinuities.

The temporal term of Equation 12 has the same form, i.e.;

$$T(i, j) = \begin{cases} C_t d(j, j') & ; \text{ if } d(j, j') < T_t \\ C_t T_t & ; \text{ if } d(j, j') \geq T_t \end{cases} , \quad (14)$$

where C_t is the cost parameter, T_t is the maximal distance for the saturation, and j' is the prediction obtained by applying ego-motion to the segmentation result of the previous cycle.

6 Experimental results

The method was tested on-line and off-line on a variety of different traffic scenarios; including downtown, highways and freeways. Figure 5(d) shows the freespace computed with the stereo image of figure 4(a). The camera captured 12 *bit* VGA greyscale images, the stereo baseline was 56 *cm*, and the focal length 1500 *px*. The carpet overlaid on the image shows the space of the road which is free of obstacles. Figure 5(c) shows the corresponding segmentation of the dynamic programming on the column/disparity grid.

A suitable application for this method is the determination of the free space while driving through road works. The lateral space available for driving might become narrow. In such a situation, the information provided by free space analysis is very valuable. Figure 6 shows two examples. The baseline of the camera system is 308 *cm*, and the images have a 12 bit greyscale VGA resolution with 820 *px* focal length. Figure 6(a) shows the free space results while driving through road works on a highway. The green carpet shows the free space detected. A prediction of the vehicle trajectory in 2 seconds is shown in blue. The walls at the left and right shows the lateral space available for driving. Figure 6(b) shows another situation on a freeway while raining. The lateral space is limited at the right by a truck, and at the left by a concrete barrier. Both are correctly detected as occupied regions, and the free space is correctly computed.

Finally, figure 6(c) shows the results of the method in a downtown scenario.

7 Conclusion

We have presented a method for the computation of free space with stochastic occupancy grids. Three types of grids were defined, their benefits and drawbacks were discussed. Applying dynamic programming to a polar occupancy grid, the optimal segmentation between free/occupied regions is found.

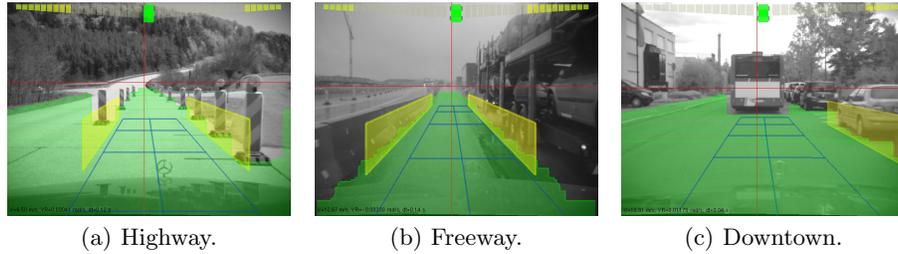


Fig. 6. Free space analysis on road works.

Acknowledgments. We would like to thank Tobi Vaudrey and Stefan Gehrig for the productive discussions. This work was partly funded by DaimlerChrysler AG and partly by the BMWi AKTIV Project Grant No. 19S6011A.

References

1. P. Cerri and P. Grisleri. Free space detection on highways using time correlation between stabilized sub-pixel precision ipm images. In *Proc. of ICRA*, 2005.
2. Alberto Elfes. Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation*, RA-3(3):249–265, June 1987.
3. U. Franke, C. Rabe, H. Badino, and S. Gehrig. 6d-vision: Fusion of stereo and motion for robust environment perception. In *DAGM '05*, Vienna, October 2005.
4. S. Kubota, T. Nakano, and Y. Okamoto. A global optimization algorithm for real-time on-board stereo obstacle detection systems. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2007.
5. M. C. Martin and H. Moravec. Robot evidence grids. Technical Report CMU-RI-TR-96-06, Robotics Institute, Carnegie Mellon University, March 1996.
6. L. Matthies. Toward stochastic modeling of obstacle detectability in passive stereo range imagery. In *Proceedings of CVPR*, Champaign, IL (USA), December 1992.
7. Larry Matthies, Takeo Kanade, and Richard Szeliski. Kalman filter-based algorithms for estimating depth from image sequences. *IJVC*, 3(3):209–238, 1989.
8. J. Miura, Y. Negishi, and Y. Shirai. Mobile robot map generation by integrating omnidirectional stereo and laser range finder. In *Proc. of IROS*, 2002.
9. Hans P. Moravec. Robot spatial perception by stereoscopic vision and 3d evidence grids. Technical Report CMU-RI-TR-96-34, Carnegie Mellon University, 1996.
10. Don Murray and James J. Little. Using real-time stereo vision for mobile robot navigation. *Autonomous Robots*, 8(2):161–171, 2000.
11. G. Sibley, L. Matthies, and G. Sukhatme. Bias reduction filter convergence for long range stereo. In *12th International Symposium of Robotics Research*, 2005.
12. A. Suppes, F. Suhling, and M. Hötter. Robust obstacle detection from stereoscopic image sequences using kalman filtering. In *23rd DAGM Symposium*, 2001.
13. Sebastian Thrun. Robot mapping: A survey. Technical report, School of Computer Science, Carnegie Mellon University, 2002.
14. Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. Intelligent Robotics and Autonomous Agents. The MIT Press, 2005.