

Stereo-based Free Space Computation in Complex Traffic Scenarios

Hernán Badino, Rudolf Mester
J. W. Goethe University
VSI Laboratory
Frankfurt am Main, Germany
hernan.badino, rudolf.mester
@vsi.cs.uni-frankfurt.de

Tobi Vaudrey
The University of Auckland
Computer Science Department
Auckland, New Zealand
t.vaudrey@auckland.ac.nz

Uwe Franke
Daimler AG
Environment Perception
Stuttgart, Germany
uwe.franke@daimler.com

Abstract

This paper presents a method for the computation of free space in complex traffic scenarios. Dynamic depth information is estimated by integrating stereo disparity images over time. Disparity and disparity speed are computed pixel-wise with Kalman filters. The stereo information is used to compute stochastic occupancy grids. Dynamic programming on a polar-like occupancy grid yields the free space. A posterior analysis of the free space allows detection of the available free corridor in front of the ego-vehicle. The method runs at 20 Hz frame rate in our vehicle.

1 Introduction

The computation of free space available in the environment is an essential task for automotive applications. The free space is the world region where navigation without collision is guaranteed. Navigable space is extremely important if an escape route is required in a critical situation. Intelligent vehicle systems need to also distinguish between moving and static objects. A static concrete wall lying in the path of a vehicle has to be treated differently than a truck moving in front of the vehicle. This paper presents a method for the computation of free space in complex traffic scenarios including moving objects.

Figure 1 shows a block diagram of the method. Stereo measurements are integrated over time obtaining disparity and disparity speed information, as well as their corresponding variances (see Section 3). This information is used to compute a stochastic occupancy grid (see Section 2). Before computing the free space, a low pass filter is applied to the occupancy grid in order to reduce the occupancy evidence noise. Free space is computed by applying dynamic programming to a polar-like representation of the occupancy grid (see Section 4). As a final step, a free space analysis is performed obtaining the free corridor in front of

the vehicle (see Section 5). The following sections address the main components of this algorithm.

2 Occupancy Grids

An occupancy grid is a two-dimensional array or grid which models occupancy evidence of the environment. Occupancy grids were first introduced in [2]. An exhaustive review on occupancy grids can be found in [12].

There are two main types of occupancy grids: deterministic and stochastic grids. Deterministic grids are basically bi-dimensional histograms counting 3D points. They are usually obtained by projecting the 3D view of measurements onto the road and counting the number of points within the same cell area ([10], [4]). The noise properties of the 3D measurements are not explicitly modeled. Grid cells with a large amount of hits are more likely to be occupied as those with none or only few points. The choice of the dis-

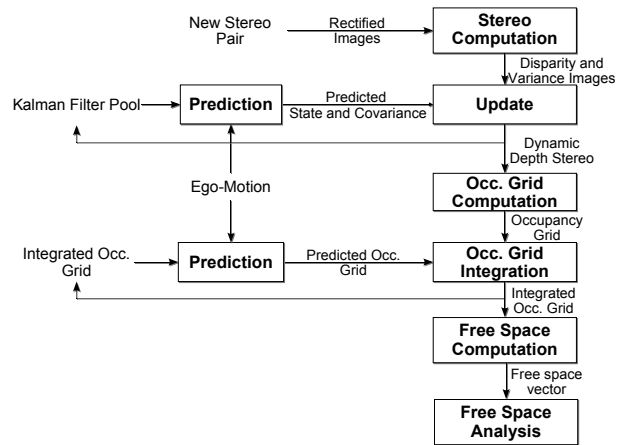


Figure 1. Block diagram of the proposed method.

cretization of cells of a deterministic grid requires the usual compromise of any sampling process. Small discretization values avoid the accumulation of points. Large discretization values lose spatial resolution, and therefore, estimation accuracy.

On the other hand, the cells of the stochastic occupancy grids maintain a likelihood or probability of occupancy. Since in a stochastic approach a single measurement affects the occupancy likelihood of the whole grid, an update function is defined. The update function specifies the operation to perform on *every cell* based on the measurement and its noise properties ([1] [11]). Stochastic occupancy grids are more expensive to compute. However, they are much more informative and do not suffer from discretization effects, as deterministic grids do.

Figure 2 shows an example of a stochastic occupancy grid. The following section describes how to compute occupancy grids from stereo measurements.

2.1 Stochastic Occupancy Grid Computation

Occupancy grids are computed by defining the function $L_{ij}(\mathbf{m}_k)$, which specifies the occupancy likelihood for cell (i, j) given the measurement \mathbf{m}_k . The measurement $\mathbf{m}_k = (u \ v \ d)^T$ is composed of image coordinates (u, v) and stereo disparity d . The measurement is the projection of some world point \mathbf{q} , such that $\mathbf{m}_k = \mathbf{P}(\mathbf{q})$, where $\mathbf{P}(\cdot)$ is the projection equation.

Given n stereo measurements, the occupancy likelihood for a cell (i, j) is obtained as

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

Assuming that cell (i, j) of a Cartesian occupancy grid is centered at world coordinate (x_{ij}, z_{ij}) the likelihood function for cell (i, j) is;

$$L_{ij}(\mathbf{m}_k) = G_{\mathbf{m}_k}(\mathbf{P}(\mathbf{p}_{ij}) - \mathbf{m}_k), \quad (2)$$

where $\mathbf{p}_{ij} = (x_{ij} \ 0 \ z_{ij})^T$, $G_{\mathbf{m}}(\boldsymbol{\xi})$ is the likelihood of obtaining an error $\boldsymbol{\xi}$ given the measurement \mathbf{m} . The error contained in the measurement \mathbf{m}_k is assumed to be Gaussian distributed. $G_{\mathbf{m}_k}$ is the multi-variate Gaussian function. Implementation details can be found in [1].

3 Pixel-Wise Estimation of Dynamic Stereo Information

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

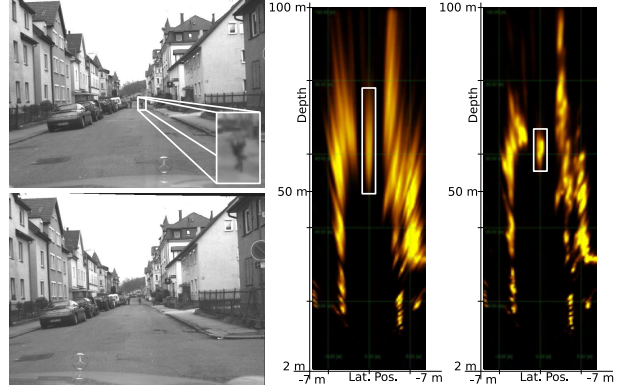


Figure 2. Stereo integration and occupancy grids. Left and right rectified images are shown at the left. The middle figure shows the evidence grid computed based only on the stereo information of the current frame. The evidence grid shown at the right was obtained with integrated disparity measurements. The occupancy likelihood is encoded as the brightness of the cells.

therefore the grouping of objects in the occupancy grid. Tracking of features in an image over time allows reduction of the position uncertainty [5]. Nevertheless, tracking is computationally expensive and highly restricts the real-time capability of a system with an increasing number of features. In [7] an iconic (pixel-based) representation for disparity estimation without tracking is introduced. The disparity image represents the state vector of a Kalman filter, under the assumption that every pixel is independent. This is equivalent to having many independent Kalman filters, one for every position of the disparity image, leading to a much more efficient computation of the filter. Nevertheless, the method assumes a static world and in complex traffic scenarios, with multiple longitudinal moving objects, this leads to a bias in the estimation of disparity [13]. In order to cope with this problem we extend this method to estimate not only disparity, but also disparity rate, i.e. the longitudinal speed of the corresponding disparity point for every pixel of the image.

This method exploits the fact that multiple disparity observations of the same point are possible thanks to the knowledge of the camera motion. Disparity measurements which are consistent over time are considered as belonging to the same spatial point, and therefore, disparity variance is reduced accordingly. By observing the change of disparity estimates over time, the simultaneous estimation of disparity rate and disparity rate variance is available. This stereo integration requires three main steps as Figure 1 shows:

- **Prediction:** The state and variance of every Kalman filter is predicted up to the current time. This is equivalent to computing expected optical flow and disparity based on ego-motion [7], while assuming constant disparity rate. The prediction of the variances includes the addition of a driving noise parameter that models the uncertainties of the system, such as ego-motion inaccuracy.
- **Measurement:** Disparity and variance images are computed based on the current left and right images. In order to maintain the real time capability of the system, a coarse-to-fine correlation-based algorithm is used [3].
- **Update:** Every filter is updated with its corresponding measurement. Current predicted disparities with no corresponding measurement (i.e. no disparity computed for 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 Kalman filter pool. Every remaining measured disparity has a corresponding estimate. If the measurement does not lie within a 3σ 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].

A detailed description of this pixel-wise estimation of stereo information can be found in [13].

An example of the improvement achieved with the iconic representation is shown in Figure 2. The occupancy grid shown at the left was obtained with standard output from the stereo algorithm while in the occupancy grid at the right was computed with an integrated disparity image. The improvement can be seen by the reduction of the *dispersion* of registered measurements. A bicyclist at approximately 50 meters away is marked over the images. The integrated disparity image shows a great improvement over the standard measured disparity image.

4 Free Space Computation

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. The set of rays must span the whole grid which leads to discretization problems. 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. For the computation of free space the first step is to transform the Cartesian grid to a polar

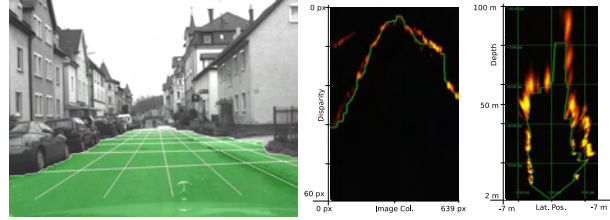


Figure 3. Free space computation. The carpet shows the computed available free space. The free space is obtained by applying dynamic programming on a Column/Disparity occupancy grid, obtained as a remapping of the Cartesian depth map, shown to the right. The resulting free space is shown as polygons on the grids.

grid by applying a remapping operation. As polar representation we use a Column/Disparity occupancy grid [1]. The middle image of Figure 3 shows an example of our Column/Disparity occupancy grid.

In the polar representation, the task is to find now the first visible obstacle in the positive direction of depth. All the space found in front of the occupied cell is considered free space. The solution to be found forms a path from left to right segmenting transversely the polar grid into two regions. Instead of thresholding each column as usually done [8] [9], dynamic programming is used. The 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 smoothness is imposed by a cost that penalizes the deviation of the current solution from a prediction.
- *Preservation of spatial and temporal discontinuities:* The truncation of the spatial and temporal costs allows the preservation of discontinuities.

Figure 3 shows an example of the free space obtained with dynamic programming.

5 Experimental results

A suitable application for this method is to determine the free space while driving through construction sites. The lateral space available for driving might become narrow. In such a situation, the information provided by free space

analysis is very valuable. Figures 4(a)-4(f) show the results on construction sites on freeways, while figures 4(g) and 4(h) show the results obtained in downtown scenarios.

All the test sequences have VGA resolution. The baseline of the stereo camera is 0.3 meters. The focal length is approximately 820 pixels. The speed of the ego-vehicle varies between 7 and 25 meters per second. The results of free space are shown as a carpet on the road. The prediction of the ego-trajectory, based on inertial sensors, is shown in front of the vehicle. The figures also show overlaid walls, which constrain the lateral free space available for driving. The positions of the walls are computed from the obtained free space. When the lateral free space in front of the ego-vehicle falls below ± 1.5 meters from the side mirrors, a wall at the corresponding position is shown.

The current implementation of the method runs at 20 Hz on a 3.2 GHz Intel Quad-Core processor.

6 Conclusion

A stereo-based method for the computation of free space was presented. Dynamic stereo information is obtained by integrating disparity images over time. This allows the calculation of accurate stochastic occupancy grids. Applying dynamic programming to a polar occupancy grid, the optimal boundary between free and occupied regions is found. A final analysis of the free space allows the robust detection of the free corridor in front of the vehicle. The experimental results show the accuracy and robustness of the method.

References

- [1] H. Badino, U. Franke, and R. Mester. Free space computation using stochastic occupancy grids and dynamic programming. In *Workshop on Dynamical Vision, ICCV*, Rio de Janeiro, Brazil, October 2007.
- [2] A. Elfes. Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation*, RA-3(3):249–265, June 1987.
- [3] U. Franke. Real-time stereo vision for urban traffic scene understanding. In *IEEE Conference on Intelligent Vehicles*, Dearborn, October 2000.
- [4] U. Franke, S. Görzig, F. Lindner, D. Mehren, and F. Paetzold. Steps towards an intelligent vision system for driver assistance in urban traffic. In *Intelligent Transportation Systems*, 1997.
- [5] 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.
- [6] L. Matthies. Toward stochastic modeling of obstacle detectability in passive stereo range imagery. In *Proceedings of CVPR*, Champaign, IL (USA), December 1992.
- [7] L. Matthies, T. Kanade, and R. 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] D. Murray and J. J. Little. Using real-time stereo vision for mobile robot navigation. *Autonomous Robots*, 8(2):161–171, 2000.
- [10] S. Nedevschi *et al.* A sensor for urban driving assistance systems based on dense stereovision. In *Intelligent Vehicles Symposium*, 2007.
- [11] A. Suppes, F. Suhling, and M. Hötter. Robust obstacle detection from stereoscopic image sequences using kalman filtering. In *23rd DAGM Symposium*, 2001.
- [12] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, 2005.
- [13] T. Vaudrey, H. Badino, and S. Gehrig. Integrating disparity images by incorporating disparity rate. In *2nd Workshop Robot Vision*, Auckland, New Zealand, February 2008. To appear.

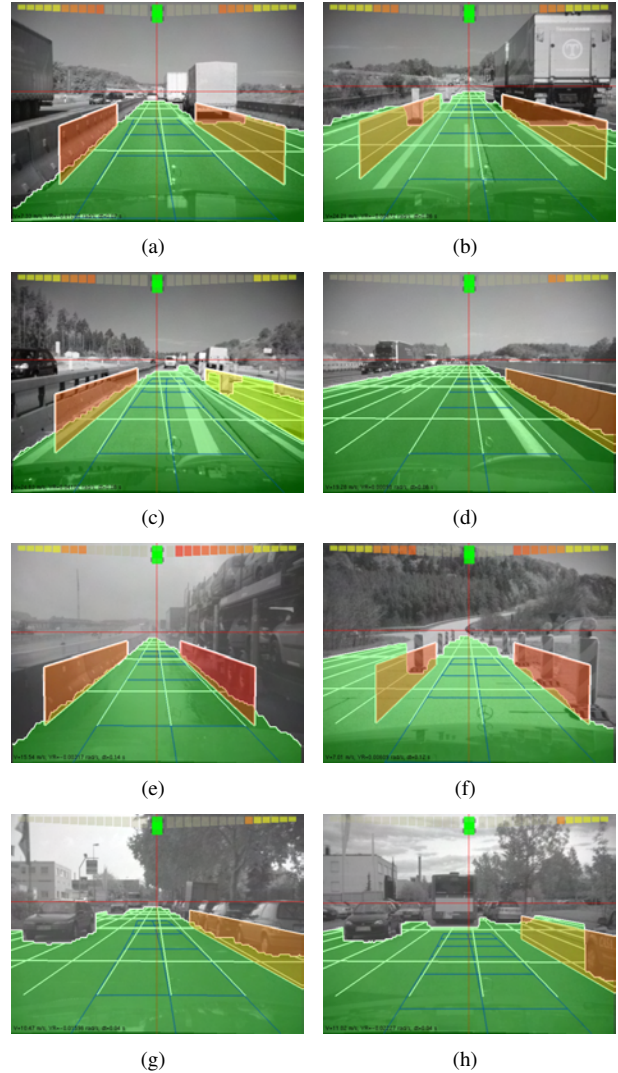


Figure 4. Experimental results in different scenarios.