https://doi.org/10.2352/ISSN.2470-1173.2018.17.AVM-282
© 2018, Society for Imaging Science and Technology

Dense Surround View Computation with Perspective Correctness

Christian Fuchs and Dietrich Paulus; Active Vision Group, University of Koblenz-Landau; Koblenz, Germany

Abstract

The observation of the direct vehicle surroundings is a critical task to both (semi-)autonomous vehicles and human drivers. A surround view from a virtual bird's view camera perspective helps to enhance operational safety in both cases. Yet, state-of-the-art methods for the computation of these views rely on planar ground plane assumptions which lead to systematic errors and highly distorted views especially on uneven ground or in off-road applications. We address this issue and propose an approach for the computation of dense perspectively correct surround views using stereo vision, a closed-surface heightmap and partial homographies. Using temporal integration of the stereo images, a high image quality is achieved.

Introduction

The observation of the direct vehicle surroundings is a safety critical task to both human drivers and (semi-)autonomous vehicles. Different types of driver assistance systems have been established to accomplish this task. Their intention is to automatically observe the vehicle surroundings or to assist the driver in navigating the vehicle and to improve operational safety.

Among different sensor modalities, such as radar, LIDAR or cameras have become of common use nowadays. The larger the vehicle gets, the more challenging the observation task gets. Especially in trucks and commercial vehicles designed for a specific task such as at construction sites, the geometry of the vehicle gets more complex.

Cameras are widely used for optical assistance systems. Applications range from simple rear driving cameras to extended environment perception using stereo vision. In case of driver assistance systems for surround views, multiple cameras are used to show the driver the direct surroundings of the vehicle. Yet, currently available systems are only suitable for urban applications when moving on a flat surface. These systems show heavy distortions as soon as the incorporated flat ground assumption is violated.

Not only obstacles such as other cars or pedestrians cause unnatural warping. When moving on uneven ground, such as in off-road or construction site environments for example, the problem raises even for non-obstacles. With a perspectively wrong view transformation, the driver may be misled by the surround views. This yields potentially dangerous situations which may affect operational safety.

Up to the best of the authors' knowledge no (published) commercial application has overcome with the issues concerning surround view assistance systems so far. In this publication, we present an approach towards the *perspectively correct* and *dense* computation of surround views. After the accumulation of environmental information in a heightmap, a closed surface is computed and used as a ground-model for the vehicle's surroundings.

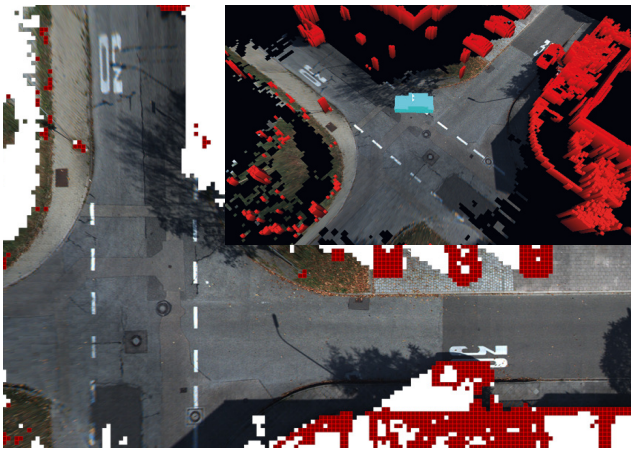


Figure 1: Dense surround view computed using the proposed method with the KITTI odometry dataset #16 [7]. The image shows the orthographic ground projection. A perspective projection of the scene, reconstructed geometry and the vehicle's position is shown on the top left. The red squares/red solids mark grid cells classified as non-ground objects.

Related Work

Different research topics must be taken into account in order to compute dense surround views. The following paragraphs summarize the most vital aspects.

Camera Geometry

When utilizing cameras as measurement instruments, the precise knowledge about the geometric property of the underlying imaging process is vital. Calibration techniques are mandatory to correctly interpret *3-D* geometry.

Tsai and Lenz [16] and Tsai [36] have published fundamental work on calibration models especially for the pinhole camera model. They introduce methods for robust camera parameter estimation using planar shapes with known geometry as reference objects. Based upon their work, Zhang [39] has proposed a refined method which is widely used. Different camera models, such as wide-angle, fisheye and catadioptric cameras need specialized calibration models. For example, Geyer and Daniilidis [8] propose an algorithm tailored towards catadioptric cameras. Scaramuzza [28] introduces an algorithm for both catadioptric and fisheye cameras which is suitable for wide-angle lenses as well. His method is used in many vehicular applications.

Surround Views and Virtual Cameras

Vehicular setups imply transformation and stitching of the original camera views as geometric limitations do not allow to capture the desired view directly. The goal is to compute a *virtual* camera's view. Perspective geometry and perspective transformations are a reasonable approach towards virtual camera views. An idea for the transformation between the views of the two cameras C_1 and C_2 is given by Hartley and Zisserman [10] and Vincent and Laganiere [37]: The view captured by C_1 shall be transformed to the view described by C_2.

Let all objects visible in C_1 be located on the same *3-D* plane. The plane can be described with at least four points on the image plane of C_1: $Q_{C_1} = \left\{ q_i \in \mathbb{R}^2 \,\middle|\, i \in \mathbb{N}^+ \right\}, |Q_{C_1}| \geq 4$

The image plane of C_1 (respectively C_2) is interpreted as a projective plane. Given corresponding image points $Q_{C_2} = \left\{ q_i \in \mathbb{R}^2 \,\middle|\, i \in \mathbb{N}^+ \right\}$ with $|Q_{C_2}| = |Q_{C_1}|$ in image coordinates of C_2, a transformation from camera C_1 to camera C_2 can be formulated using a homography matrix $H_{C_1 \to C_2} \in \mathbb{P}^{2\times 2}$. Matrix $H_{C_1 \to C_2}$ is considered constant, assuming the cameras to have fixed lenses and a rigid affine transformation between their poses.

Homographies are a good starting point for surround view computations. They are commonly used for the warping process. They include a a flat world assumption, which seems reasonable on a first look for vehicles, as the street can be assumed to be a plane.

Liu, Kin and Chen [19] use cameras positioned around the vehicle and utilize homography matrices to transform the images and finally stitch the images to a surround view. They warp the resulting images to a fisheye view to create a more natural look of the results. An integration of a homography based approach with a hardware setup is proposed by Luo *et al.* [21]. Thomas *et al.* [35] focus on cost-efficient hardware and stitch top view images on it. Sato *et al.* [27] utilize fish-eye cameras together with homographies on spatio-temporal data, whereas Li and Hai [18] focus on the calibration of a multi-view bird's eye view.

However, these methods show heavy artifacts for objects violating the assumptions (of the homography). When the vehicles encounter objects, which do not fulfill the planar constraint, artefacts show a vanishing-point-like behavior. Yet, they are used in bird's view systems nowadays. The overall *homography shadowing effect* is described by Fuchs and Paulus [4, 5]. The authors propose a first approach towards perspectively correct virtual bird's views. They use stereo cameras in combination with disparity matching to compute point-based *3-D* geometry and project the resulting point cloud perspectively correct to the ground plane. However, the approach shows sparse data in areas with overlapping geometry. This publication is intended to overcome with this issue.

Stereo Vision, Datasets and Visual Odometry

Multiple view geometry is the key to extract *3-D* information from camera images. The use of multiple camera setups can be used to compute depth information from multiple images. Hartley and Zisserman [10] have published fundamental work on the topic and summarize the principle behind this approach.

Known camera intrinsics and relative positions and orientations between the cameras are a prerequisite for depth estimation and can be determined using calibration techniques. Given synchronously grabbed camera frames, the correspondence problem has to be solved in order to compute disparities.

Existing correspondence matching approaches classify into two main types: On the one hand, keypoint-based approaches use image feature algorithms like [17], [20] or [26]. Using adequate distance measurements, features are matched in between the synchronous frames for correspondence assignment. For example, Grimson [9] uses image features in order to find stereo correspondences. Horaud and Skordas [13] group features first in order to extract correspondences. Results of keypoint- and/or image feature-based algorithms usually show sparse *3-D* data with high accuracy.

On the other hand, block-matching can be utilized for correspondence matching. Various algorithms and improvements have been developed and published so far. Hirschmüller *et al.* [12] use mutual information and pixel-wise matching in their *semiglobal matching (SGM)*. The algorithm has become popular and has already been adapted to particular scenarios, for example for in-vehicle applications [31, 11]. Einecke and Eggert [3] utilize a local correspondence approach and significantly reduce the SGMs execution while maintaining correspondence quality. Results of block-matching algorithms tend to show dens disparity data with slightly minor depth quality in comparison to keypoint-based approaches. Many publications and algorithms for stereo disparity estimation have already been presented. This work does not focus on this issue. We use the algorithm proposed by Hirschmüller *et al.* [12].

Of course, a lot of approaches towards stereo processing in vehicular environments (e. g. [24], [1], [14] and many more) have been published. Yet, no publication addresses perspectively surround views so far.

Fuchs and Paulus [4] use stereo vision and transform between camera views to create a perspectively correct surround view. The approach shows sparse result images while the perspective correctness is maintained. Several stereo datasets recorded for vehicular applications in particular have been published [25, 29, 30]. Geiger, Lenz and Uratsum [7] present the *KITTI Stereo Benchmark*, which has become popular in recent works. It contains datasets for different purposes. The odometry benchmark datasets from their collection are used in our work.

The precise knowledge about the movement between to frames is vital for temporal integration of point clouds. When (stereo) cameras are used as sensors, visual odometry can be utilized for movement and relative position and orientation estimation. Cvišić and Petrović [2] propose a stereo visual odometry system which is currently amongst the best-ranking algorithms on the KITTI odometry benchmark [7]. We follow the algorithm by Cvišić and Petrović [2] in this publication.

Image-based Rendering

Methods of image-based rendering can be used to create virtual views, too. An overview of different algorithms is given by Shum and Kang [32]. They group existing approaches by the geometric modelling strategy used. Surround view computation is targeted towards real-time processing so that complex precomputed or pre-modeled knowledge cannot be utilized. Only implicit geometry methods are reasonable in this context. Laveau and Faugeras [15] predict views using the fundamental matrix and two images. Zinger, Do and De With [42] discuss a free-viewpoint depth based rendering for *3-D*-TV applications. Their

method relies on disparity maps. Vogt *et al.* [38] use light-fields to improve the quality in image sequences.

However, these publications discuss the computation of virtual camera views with camera poses close to the original camera's views, e. g. light positions shift and/or light rotation. In case of vehicular surround views, extensive shifts and rotations become necessary. Yet, the publications contain fundamental work on the issue, but do not present proper solutions for surround views.

Heightmaps and Heightmap Texturing

Two commonly used data structures for the representation of closed surface are of common use: Heightmaps and variants of the voxel-based truncated signed-distance function (TSDF) [6].

Especially in the context of SLAM, heightmaps are used widely. Heightmaps are optimization simultaneously to the position and orientation of the camera: Various authors use heightmaps as their map data structure in a SLAM context [7, 3, 8]. They simultaneously optimize the geometry of the reconstructed heightmap and the position and orientation of the camera.

Sugimoto, Kotooka and Okutomi [33] use homographic mapping between subsequent stereo frames and formulate a cost function on the heightmap geometry. Yet, they do not explicitly texture the surface. Zienkiewicz, Davison and Leutenegger [40] and Zienkiewicz et al. [41] obtain their cost function by quantifying the difference between the reprojection of the heightmap's depth and the output of a motion stereo algorithm. They focus on online optimization rather than high-resolution texturing. The authors mainly use color information for visualization purposes.

Motooka et al. [22] optimize the texture of the heightmap across a large number of camera images photometrically. While the results show impressive quality, the authors note that their optimization method is currently not fast enough for online operation. Tanner et al. [34] use TSDFs to map large areas and show a relatively detailed and colored mesh as result of their reconstruction. However, they focus on the ability to map large areas rather than texture details: Colorization only occurs on a per-vertex basis, which limits the texture resolution to their voxel size of 10 cm.

A *3-D* reconstruction approach is presented by Gallup, Frahm and Pollefeys [6]. They cluster depth images into voxels and fit them into a heightmap while focusing on a continuous surface at the cost of texturing quality.

Shifting Grid Map

To model the vehicle's environment, we use a *2-D* grid map as the basic geometric environment representation. The grid map is oriented on the ground plane below the vehicle. Its cells are equidistant and its bases are orthogonal. The reference coordinate system is the world coordinate system.

Figure 2 shows the grid map in relation to the vehicle and the world coordinate system: The origin of the world coordinate system 0^{w} defines the origin of the grid map.

The grid map G is aligned to the world coordinate system's xy-plane (ground plane) and has equidistant cells with orthonormal axes. Parameter $g \in \mathbb{R}$ defines the side length of a single grid cell. The positions of the cells are and remain constant to the initially defined world coordinate systems. The hereby achieved discrete sampling of the ground plane enables an indexed access to the grid cells. Each cell can be addressed using *2-D*-coordinate

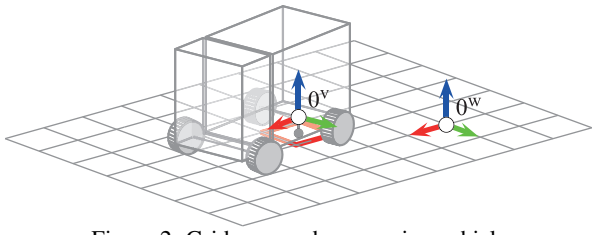


Figure 2: Grid map under a moving vehicle

$(c_x, c_y) \in \mathbb{Z}^2$. The bounds of the grid are assumed to be infinite for modelling purposes.

Position and orientation of vehicles and reference systems are relevant for any investigation regarding surround views. To create an easier readability, both components can be combined to a so called pose. As a first component, a pose contains the position in form of a translation vector. As a second component, the rotation of the object is expressed as a unit quaternion in the space of a unit-3-sphere in *4-D* Euclidean space. The space $\mathbb{S}^3$ is a subspace of the Hamilton space for unit quaternions in general ($\mathbb{S}^3 \subset \mathbb{H}$). Special operations can be used for the combination of poses. For the following, the operator $\oplus$ describes a forward composition of two poses.

The position and orientation of the vehicle in the world coordinate system is described by pose $\mathfrak{v} \in \mathbb{R}^3 \times \mathbb{S}^3$ with $\mathfrak{v} = \langle q, \phi \rangle$. At this point, only the position component of pose $\mathfrak{v}$ is relevant. As the vehicle is moving on the ground it can be projected to the grid map. With $q \in \mathbb{R}^3$ the vehicle's position in the world coordinate system, the *2-D* position on the grid map is described by function $\rho_G : \mathbb{R}^3 \to \mathbb{R}^2$:

$$\rho_G(q) = \rho_G\left((q_x, q_y, q_z)^{\mathrm{T}}\right) = (q_x, q_y)^{\mathrm{T}}$$

This projected *2-D* position is associated with a corresponding grid cell. Function $\gamma : \mathbb{R}^2 \to \mathbb{Z}^2$ maps the *2-D* vehicle position to the rasterized ground grid. This yields for the mapping of a vehicle position to a grid cell:

$$c_p = \gamma(\rho_G(q))$$

As the goal is to model and visualize the surroundings of the vehicle, we of course do not use an infinite grid map. Instead, we use as *local* grid map which is centered around the vehicle. The map however remains in the grid layout with world reference. While the vehicle moves (relatively to the world coordinate system), the center cell has to be chosen adequately and the map has to adapt to it.

The local grid map around the vehicle is aligned to the center cell and its extensions are defined by parameter $e \in \mathbb{N}$ which defines the number of cells in each direction. The local grid contains $(2 \cdot e + 1)^2$ cells, with *local* cell indexations in the range of $([-e; e], [-e; e]) \in \mathbb{Z}^2$. This yields a quadratic excerpt of the world grid which is a adequate choice for surround view applications. Depending on the application, a different geometric positioning of the center cell and the extents may be a proper choice.

Let $t_\tau \in \mathbb{R}^2$ be the *2-D* grid reference position of $\mathfrak{v}_\tau$ at time step τ and $t_{\tau+1} \in \mathbb{R}^2$ the position at time step $\tau + 1$. An update of the shifting grid's bounds regarding the world grid is necessary, if $\gamma(t_\tau) \neq \gamma(t_{\tau+1})$. The principle behind a grid shift is shown in Figure 3.

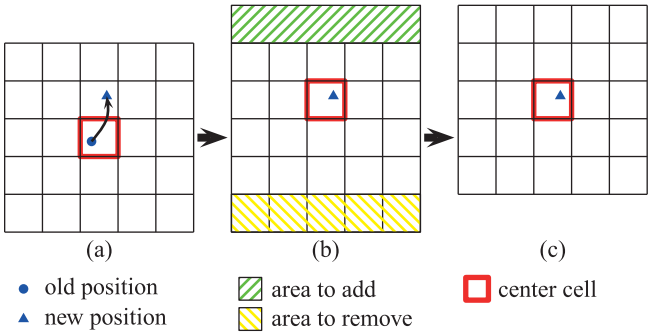


Figure 3: Principle of grid shifting. As long as the vehicle's *2-D* ground position remains in the center cell, the grid is not shifted. As soon as the vehicle moves to another grid cell (a), the center cell is moved to the cell the vehicle moved to (b). At the same time, the candidates for the grit shift are selected (b). The grid's borders are then resized so that the center constraint holds for the center cell (c). Grid contents remain during the resizing.

The grid map is used to store environmental information. Therefore, the cells are a discretization of the surroundings with a fixed relation to the world reference. This enables accumulative data collection while the vehicle is moving.

We use stereo cameras and/or a *3-D* laser scanner for geometric sampling. This means that – irrelevant of the sensor modality – point clouds are used in first stage. In case of the stereo camera, disparity algorithms as explained above are used for depth information. However, the raw output of a *3-D* laser scanner can be used directly.

For explanation, we only formulate our approach with one sensor. Of course, multiple sensors can be equipped and used as input for the method proposed. The sensor coordinate system s has its origin in the sensor mounted on the vehicle. An overview of the coordinate systems is given in Figure 4.

The set of points A^{s}_{τ} at timestep τ in the coordinate system of the sensor s is defined as:

$$A^{\mathrm{s}}_{\tau} = \left\{ p \,\middle|\, \forall\, p \in \mathbb{R}^3 \right\}$$

In order to link the point set to grid map cells, their coordinates must be transformed first. Given the *6-D* pose $\mathfrak{v} \in \mathbb{R}^3 \times \mathbb{S}^3$ which defines the relative pose of the vehicle regarding the world coordinate system. The position component q can be mapped to a grid cell in G using function γ. Yet, the grid cell's origin is necessary for the transformation. Let c be the corresponding cell. The origin 0^{c} of the cell c is given by function $\omega : \mathbb{R}^3 \to \mathbb{R}^3$:

$$\omega(q) = \left(\left\lfloor q_x \cdot g^{-1} \right\rfloor \cdot g, \left\lfloor q_y \cdot g^{-1} \right\rfloor \cdot g, 0 \right)^{\mathrm{T}}$$

As there is no rotation in the coordinates system's axes between the world coordinate system and a grid cell's coordinate system, the rotation quaternion of pose $\mathfrak{v}$ does not imply further transformation. The vehicle pose $\mathfrak{v}$ in the grid cell's coordinate system c is:

$$\mathfrak{v}^{\mathrm{c}} = \langle q^{\mathrm{w}} - \omega(q^{\mathrm{w}}), \phi^{\mathrm{w}} \rangle$$

Pose $\mathfrak{s}$ defines the transformation from the vehicle coordinate system to the sensor data's origin (in case of Figure 4 the camera center of left camera of the stereo system). Given center cell c of grid G, the accumulated transformation pose $\mathfrak{a}$ to register a point cloud A^{s}_{τ} is:

$$\mathfrak{a} = \mathfrak{v}^{\mathrm{c}} \oplus \mathfrak{s}$$

Let function $\zeta : \mathbb{R}^3 \times \mathbb{S}^3 \to \mathbb{P}^{3\times 3}$ represent the affine transformation matrix applying the transform of a pose. The computation of A^{c}_{τ} yields:

$$A^{\mathrm{c}}_{\tau} = \{ \zeta(\mathfrak{a}) \cdot \tilde{p} \,|\, \forall p \in A^{\mathrm{s}}_{\tau} \}$$

To cluster the points for fusion in the grid cells, a set $C_{\tau;u,v} \subseteq A^{\mathrm{c}}_{\tau}$ is defined for each cell:

$$C_{\tau;u,v} = \left\{ p \,\middle|\, p \in A^{\mathrm{c}}_{\tau} \wedge \gamma(\omega(p)) = (u,v)^{\mathrm{T}} \right\}$$

The shifting grid map G containing the accumulated *3-D* points of the environment is the basis for a *closed surface model* of the terrain. We follow a heuristic approach which is close to the one in [23] to estimate each cell's height: Given the points in A^{c}_{τ} and their corresponding distances to the sensor's origin in D_{τ} (with $D_{\tau;u,v}$ analogue):

$$D_{\tau} = \{ d \,|\, \forall p \in A^{\mathrm{s}}_{\tau}, d = \|p\| \}$$

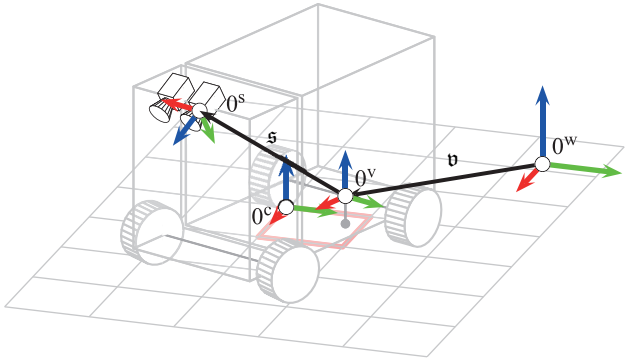


Figure 4: Coordinate systems for world (w), vehicle (v), center cell (c) and sensor (s). Pose $\mathfrak{v}$ transforms from world the vehicle coordinate system. The sensor pose $\mathfrak{s}$ defines the coordinate system of the sensor data within the vehicle.

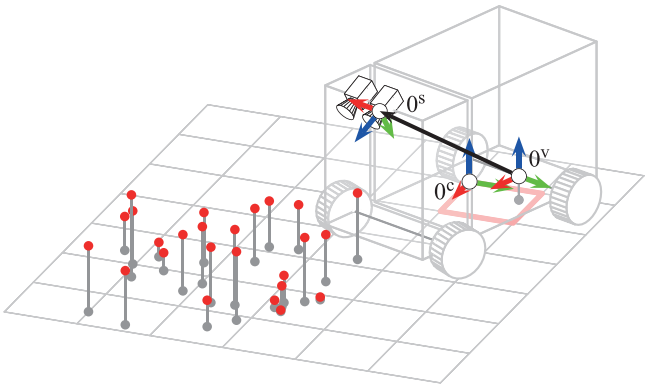


Figure 5: Matching of point clouds to the grid structure. Each point gets assigned to the containing cell using the *2-D* projection of each point.

Algorithm 1 Cell height estimation (following [23])

$\sum_h \leftarrow 0 \qquad \sum_w \leftarrow 0 \qquad \hat{z} \leftarrow 0$
$\Gamma \leftarrow \{(z,d) \,|\, p \in C_{u,v},\ d \in D_{u,v},\ z = p_z,\ p \text{ and } d \text{ associated}\}$
$n \leftarrow |\Gamma|$
sort tuples in Γ **ascending by** z
for $i := 1$ **to** n **do**
  $h \leftarrow \Gamma[i].z$ ▷ height component of point
  $w \leftarrow \chi(\Gamma[i].d)$ ▷ weight factor
  if $i > 0.5 \cdot n$ **and** $h - \hat{h} > k$ **then** ▷ k a ground threshold
    break
  $\sum_h \leftarrow \sum_h + w_i \cdot h_i$
  $\sum_w \leftarrow \sum_w + w_i$
  $\hat{z} \leftarrow \sum_h \cdot \sum_w{}^{-1}$
return $\hat{z}$

Using $C_{\tau;u,v}$ and the corresponding distances in $D_{\tau;u,v}$, the points are then added to the correct cells around the vehicle. The cells are used to accumulate points over various timesteps. This way, an *update and refinement* of each cell is possible:

$$C_{u,v} := C_{u,v} \cup C_{\tau;u,v} \qquad D_{u,v} := D_{u,v} \cup D_{\tau;u,v}$$

The distance to the sensor is regarded a quality criterion for the *3-D* point. The closer the point to the sensor, the more reliable it is in terms of height estimation. The distance is therefore transformed into a weight using function $\chi : \mathbb{R} \to \mathbb{R}$:

$$\chi(d) = \left(1 + \frac{d}{8}\right)^{-0.5}$$

The iterative algorithm for cell height estimation based on the accumulated point cloud data is depicted in Algorithm 1. Cells with a high point height variance are regarded to be non-ground obstacle cells which would violate the closed surface approach. Therefore, they are modelled as solids in the size of the grid cell. Let the results of the algorithm be accessible through $\xi : \mathbb{Z}^2 \to \mathbb{R}$ as a mapping from cell coordinates to the computed heights.

Using the heights in ξ, the closed surface is computed: We use neighbored cells, to compute a mean height for the surface vertex at the cell's corners. This way, an equidistant orthonormal set of points is created. The closed surface coordinate at the origin of cell (u,v) is expressed by function $\beta : \mathbb{Z}^2 \to \mathbb{R}^3$:

$$\beta(u,v) = \left(g \cdot u,\ \ g \cdot v,\ \ \frac{1}{4} \sum_{i=-1}^{0} \sum_{j=-1}^{0} \xi(u+i, v+j)\right)^{\mathrm{T}}$$

Each (quadratic) cell (u,v) in grid map G is over-spanned with two triangles:

$$\begin{aligned} \triangle_1(u,v) &= \{\beta(u,v),\ \beta(u+1,v),\ \beta(u,v+1)\} \\ \triangle_2(u,v) &= \{\beta(u+1,v+1),\ \beta(u,v+1),\ \beta(u+1,v)\} \end{aligned}$$

The final closed surface heightmap consists of all triangles which over-span the grid map. Figure 6 visualized the principle behind the computation. The smoothing effect achieved due to the averaging is clearly visible there.

Partial Homography Warping

One of the main problems regarding state of the art methods for surround view generation is the unnatural warping of non-ground objects. Yet, in image regions where the ground plane assumption is fulfilled, correct transformations to the virtual view are shown. Using the closed surface heightmap of the environment, partially plane parts are created. The geometric structure of the heightmap is updated continuously while the car is moving.

As a heightmap alone does not yet help to visualize a surrounding view, camera images are needed in the next step. These camera images can be taken by a stereo system which can be used as input for the heightmap computation as well. It is also possible to use any other and/or multiple cameras on the vehicle as an image source. However, the following paragraphs focus on a single camera as input to emphasize the methodology.

Given a camera with known intrinsic matrix $K \in \mathbb{P}^{2\times 2}$ and pose $\mathfrak{k} \in \mathbb{R}^3 \times \mathbb{S}^3$ in the vehicle. (Of course, lens distortions are vital, too. For simplification issues, they are not discussed in the following, yet have to be taken care of.) The idea is to partially warp the camera image to the heightmap's triangles (while maintaining a visibility constraint).

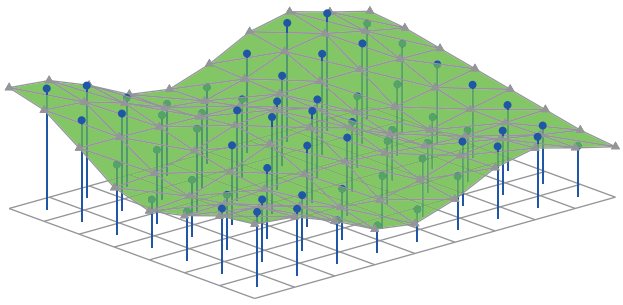
A planar texture image T is used to accumulate each triangle's image content. The texture image's dimensions are a multiple of the cells in the grid map. This way, each *cell* gets assigned a texture resolution $r \in \mathbb{N}$, resulting in a texture region of r by r pixels per cell. The dimensions of the texture for the whole grid map are $r \cdot (2 \cdot e + 1)$ by $r \cdot (2 \cdot e + 1)$ pixels.

Given the *3-D* coordinates of a triangle in the closed surface heightmap, the projection onto the image plane can be computed with the intrinsic matrix K. At first, the points have to be transformed to the camera's coordinate system k using the accumulated pose $\mathfrak{v}^{\mathrm{c}} \oplus \mathfrak{k}$ and the cell coordinates. Given a triangle:

$$\triangle^{\mathrm{k}} = \left\{p_1^{\mathrm{k}}, p_2^{\mathrm{k}}, p_3^{\mathrm{k}}\right\}$$

To formulate a point correspondence problem for homography computation, at least four points are necessary. We achieve this by adding a fourth virtual point on the triangle plane using a linear combination of the triangle points:

$$p_4^{\mathrm{k}} = p_2^{\mathrm{k}} + p_3^{\mathrm{k}} - p_1^{\mathrm{k}} \qquad \square^{\mathrm{k}} = \triangle^{\mathrm{k}} \cup \left\{p_4^{\mathrm{k}}\right\}$$



- cell heights as computed by Algorithm 1
- closed surface heights/corners (β)
- planar closed surface element ($\triangle_1/\triangle_2$)

Figure 6: Closed surface heightmap. The cell heights are used for generating the closed surface heightmap which is partially planar.

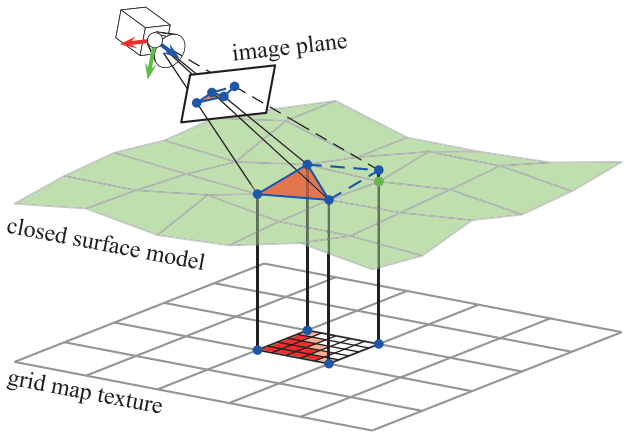


Figure 7: Principle of partial heightmap texturing. The *3-D* points are projected to the image plane of the camera to get the bounds of the corresponding image region. The image region is then warped onto the grid map texture plane.

The projection onto the *2-D* image plane of the camera (coordinate system i) is computed by multiplying with K:

$$\square^{\mathrm{i}} = K \cdot \square^{\mathrm{k}} = \left\{ K \cdot p_1^{\mathrm{k}}, K \cdot p_2^{\mathrm{k}}, K \cdot p_3^{\mathrm{k}} \right\}$$

The *2-D* point correspondences in the target texture are given the corresponding cell's texture coordinates in pixels as explained above. The homography matrix $H \in \mathbb{P}^{2\times 2}$ can be computes using methods as explained in [10]. To apply the homography matrix, a point is multiplied with it. To get the correct *2-D* coordinate, a perspective division is necessary. Both steps are accumulated in function $h : \mathbb{R}^2 \to \mathbb{R}^2$:

$$h(t) = \left(y_x \cdot {y_z}^{-1},\ y_y \cdot {y_z}^{-1} \right)^{\mathrm{T}} \quad \text{with} \quad y = \left(y_x, y_y, y_z \right)^{\mathrm{T}} = H \cdot \tilde{t}$$

Of course, the warped image can be applied only to the part of the cell's texture which belongs to the current triangle. As the interpolated fourth point causes a texture excerpt for the whole grid cell, the correct half for the triangle must be selected. At the diagonal where the two textures need, blending is applied. The texture image is accumulated respectively updated by the transformation of all patches $\square_v^{\mathrm{k}}$:

$$T := \left\{ h\left(\square_v^{\mathrm{k}} \right) \,\middle|\, \forall \square_v^{\mathrm{k}} \in G \right\}$$

The partial homography warping process is depicted in Figure 7. Each triangle patch is written to a texture layer which is then mapped to the closed surface mesh. Of course, a visibility constraint must be maintained so that no projections to areas behind obstacles are computed.

As a result, both the perspectively correct surround view in an orthographic projection (in the texture layer) as well as *3-D* perspective views of the scenery can be generated.

Test Results and Conclusion

The KITTI odometry datasets [7] is used to test the proposed method. We choose this publicly available dataset to enable comparison with future methods. Although the dataset only includes

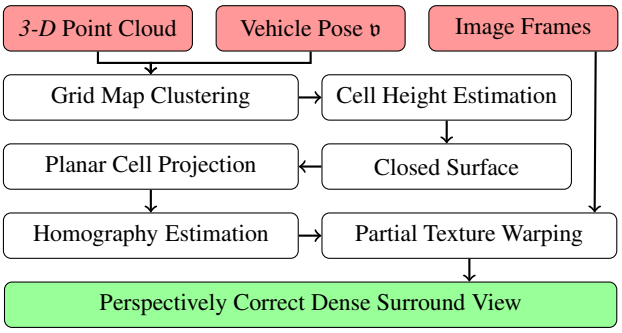


Figure 8: Overview of the system's components. Red boxes are input for the proposed method, the green box indicates the output.

camera views in driving direction (instead of cameras distributed around the vehicle) it is well suitable for our method.

At first, a visual odometry algorithm [2] is used to compute the relative poses between the frames. The algorithm by Hirschmüller [12] is utilized for disparity estimation. The component setup is depicted in Figure 8 and shows how the single steps as explained above are connected.

For the results presented, the grid map was initialized with a cell side length of $g = 3.5^{-1}$ ($\approx 29\,\mathrm{cm}$ in the world). The grid's extents are configured to 61 by 61 cells ($e = 30$). Texture resolution is set to $r = 16\,\mathrm{px}$ (each pixel will represent a patch of $1.7\,\mathrm{cm}$ by $1.7\,\mathrm{cm}$ in the world).

Figures 1, 9, 10 and 11 show results from different datasets. The temporal integration of the *3-D*-data allows a precise modelling of the vehicle surroundings and the detection of off-ground obstacles.

In case of Figure 9, the image quality is visible, e. g. at the cobblestones. Off-ground obstacles such as fences and parking cars are detected correctly. While Figure 10 shows a road passing through an uneven grassland, the terrain surrounding the street in Figure 11 is much steeper. Details like manholes and tar patches on the road are clearly presented.

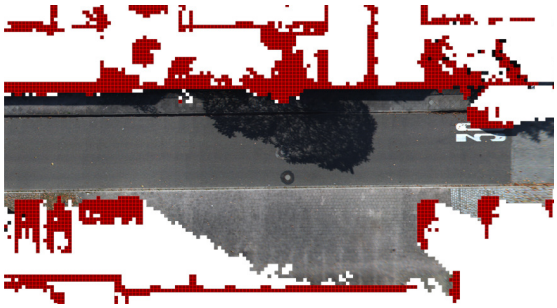
The approach shows good image quality while maintaining perspective correctness. The temporal integration of image data in the showcase of course results in "old" image data being used for the surround view (e. g. behind the camera) and is due to the dataset used. Yet, there is no dataset publicly available with cameras distributed around the vehicle. In means of a real-world application, live data for the direct vehicle surroundings is mandatory. However, the principle behind the proposed approach can be shown with the datasets used. Using cameras distributed around the vehicles for partial homography mapping will overcome this problem. Additionally, visualizations marking that the image at a specific location is from a former view could be used as well.

For future work, it is planned to enhance the obstacle detection and to subdivide grid cells in geometrically complex areas for a better approximation of the ground.

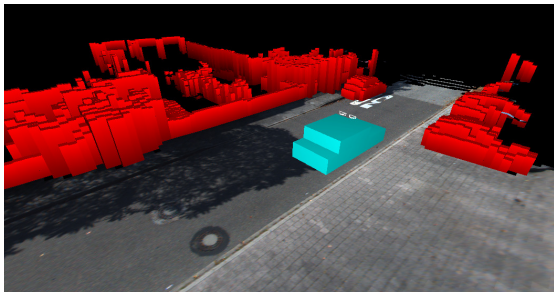
References

[1] A. Broggi, C. Caraffi, R. I. Fedriga, P. Grisleri, and I. Parma. Obstacle Detection with Stereo Vision for Off-Road Vehicle Navigation. In *IEEE Conference on Computer Vision and Pattern Recognition*, page 65, San Diego, 2005.

[2] I. Cvišić and I. Petrović. Stereo odometry based on careful feature



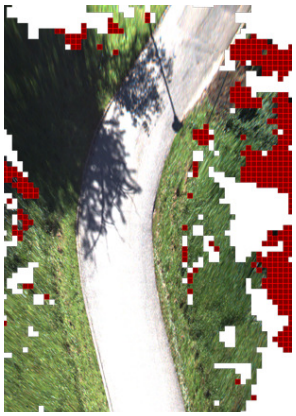
(a) Ground Texture (orthographic).
Red patches mark cells classified as obstacles.



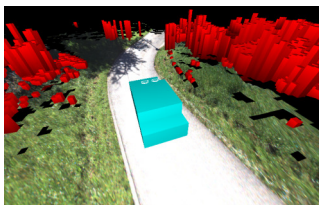
(b) *3-D* View with Car Pose.
The red solids mark cells classified as obstacles and are visualized using the estimated obstacle heights.

(c) Scene image (KITTI odometry dataset #16, frame 150 [7]))
The two frustums above the vehicle mark the stereo camera's position.

Figure 9: Perspectively correct dense surround view result using KITTI odometry dataset #16.



(a) Ground Texture (orthographic)

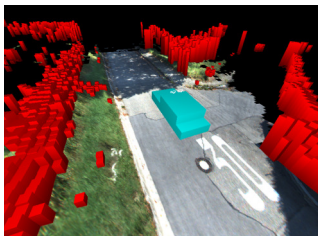


(b) *3-D* View with Car Pose

(c) Scene image (KITTI odometry dataset #10, frame 33 [7])

Figure 10: Perspectively correct dense surround view result using KITTI odometry dataset #10

(a) Ground Texture (orthographic)



(b) *3-D* View with Car Pose

(c) Scene image (KITTI odometry dataset #3, frame 503 [7])

Figure 11: Perspectively correct dense surround view result using KITTI odometry dataset #3

selection and tracking. In *2015 European Conference on Mobile Robots, ECMR 2015 - Proceedings*, 2015.

[3] N. Einecke and J. Eggert. A Multi-Block-Matching Approach for Stereo. In *IEEE Intelligent Vehicles Symposium, Proceedings*, pages 585–592, 2015.

[4] C. Fuchs and D. Paulus. Perspectively Correct Bird's Views Using Stereo Vision. In *Autonomous Vehicles and Machines Conference, IS&T Electronic Imaging 2017*. IS&T Digital Library, 2017.

[5] C. Fuchs and D. Paulus. Perspectively correct construction of virtual views. In *Proceedings of the 6th International Conference on Pattern Recognition Applications and Methods - Volume 1: ICPRAM*, pages 626–632. Scitepress, 2017.

[6] D. Gallup, J.-M. Frahm, and M. Pollefeys. A Heightmap Model for Efficient 3D Reconstruction from Street-Level Video. *Int. Conf. on 3D Data Processing, Visualization and Transmission*, 6, 2010.

[7] 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*, pages 3354–3361, 2012.

[8] C. Geyer and K. Daniilidis. Catadioptric projective geometry. *International Journal of Computer Vision*, 45(3):223–243, 2001.

[9] W. E. Grimson. Computational experiments with a feature based stereo algorithm. *IEEE transactions on pattern analysis and machine intelligence*, 7(1):17–34, 1985.

[10] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition, 2003.

[11] S. Hermann and R. Klette. Iterative semi-global matching for robust driver assistance systems. In *Asian Conference on Computer Vision*, pages 465–478, 2013.

[12] H. Hirschmüller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341, 2008.

[13] R. Horaud and T. Skordas. Stereo correspondence through feature grouping and maximal cliques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(11):1168–1180, 1989.

[14] C. G. Keller, C. Hermes, and D. M. Gavrila. Pattern Recognition: 33rd DAGM Symposium, Frankfurt/Main, Germany, August 31 – September 2, 2011. Proceedings. In R. Mester and M. Felsberg, editors, *DAGM*, chapter Will the P, pages 386–395. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[15] S. Laveau and O. Faugeras. 3-D scene representation as a collection of images. *Proceedings of 12th International Conference on Pattern Recognition*, 1, 1994.

[16] R. K. Lenz and R. Y. Tsai. Techniques for calibration of the scale factor and image center for high accuracy 3-D machine vision metrology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(5):713–720, 1988.

[17] S. Leutenegger, M. Chli, and R. Y. Siegwart. BRISK: Binary robust invariant scalable keypoints. In *2011 IEEE International Conference on Computer Vision (ICCV)*, pages 2548–2555, 2011.

[18] S. Li and Y. Hai. Easy calibration of a blind-spot-free fisheye camera system using a scene of a parking space. *IEEE Transactions on Intelligent Transportation Systems*, 12(1):232–242, 2011.

[19] Y. C. Liu, K. Y. Lin, and Y. S. Chen. Bird's-eye view vision system for vehicle surrounding monitoring. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4931 LNCS:207–218, 2008.

[20] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[21] L. Luo, I. Koh, S. Park, R. Ahn, and J. Chong. A software-hardware cooperative implementation of bird's-eye view system for camera-on-vehicle. *2009 IEEE International Conference on Network Infrastructure and Digital Content*, pages 963–967, 2009.

[22] K. Motooka, S. Sugimoto, M. Okutomi, and T. Shima. 360-Degree 3D Ground Surface Reconstruction Using a Single Rotating Camera *. In *Proceedings of 7th Workshop on Planning, Perception and Navigation for Intelligent Vehicles (PPNIV2015)*, pages 147–152, 2015.

[23] F. Neuhaus, N. Wojke, C. Winkens, B. Krayer, D. Paulus, and M. Häselich. Autonomous 3d Terrain Mapping and Object Localization for the Spacebot Camp 2015. In *International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, 2016.

[24] D. Pfeiffer and U. Franke. Towards a Global Optimal Multi-Layer Stixel Representation of Dense 3D Data. *Procedings of the British Machine Vision Conference 2011*, pages 51.1–51.12, 2011.

[25] D. Pfeiffer, S. Gehrig, and N. Schneider. Exploiting the power of stereo confidences. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 297–304, 2013.

[26] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB - an efficient alternative to SIFT or SURF. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2564–2571, 2011.

[27] T. Sato, A. Moro, A. Sugahara, T. Tasaki, A. Yamashita, and H. Asama. Spatio-temporal bird's-eye view images using multiple fish-eye cameras. *Proceedings of the 2013 IEEE/SICE International Symposium on System Integration*, pages 753–758, 2013.

[28] D. Scaramuzza. *Omnidirectional vision: from calibration to robot motion estimation*. PhD thesis, ETH Zürich, 2007.

[29] T. Scharwächter, M. Enzweiler, U. Franke, and S. Roth. Efficient Multi-Cue Scene Segmentation. *German Conference on Pattern Recognition (GCPR)*, 2013.

[30] T. Scharwächter, M. Enzweiler, S. Roth, and U. Franke. Stixmantics: a medium-level model for real-time semantic scene understanding. In *Computer Vision ECCV 2014*, pages 533–548. Springer International Publishing, 2014.

[31] T. Scharwachter, M. Schuler, and U. Franke. Visual guard rail detection for advanced highway assistance systems. *IEEE Intelligent Vehicles Symposium, Proceedings*, (Iv):900–905, 2014.

[32] H.-Y. Shum and S. B. Kang. A review of image-based rendering techniques. *Proc. SPIE Visual Communications and Image Processing*, pages 2–13, 2000.

[33] S. Sugimoto, K. Motooka, and M. Okutomi. Direct generation of regular-grid ground surface map from in-vehicle stereo image sequences. *Proceedings of the IEEE International Conference on Computer Vision*, pages 600–607, 2013.

[34] M. Tanner, P. Pinies, L. M. Paz, and P. Newman. DENSER Cities: A System for Dense Efficient Reconstructions of Cities. *arXiv preprint arXiv:1604.03734*, 2016.

[35] B. Thomas, R. Chithambaran, Y. Picard, and C. Cougnard. Development of a cost effective bird's eye view parking assistance system. *2011 IEEE Recent Advances in Intelligent Computational Systems*, pages 461–466, 2011.

[36] R. Y. Tsai. A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses. *IEEE Journal on Robotics and Automation*, 3(4):323–344, 1987.

[37] E. Vincent and R. Laganière. Detecting planar homographies in an image pair. *2nd International Symposium on Image and Signal Processing and Analysis*, 0(2):182–187, 2001.

[38] F. Vogt, S. Krüger, J. Schmidt, D. Paulus, H. Niemann, W. Hohenberger, and C. H. Schick. Light fields for minimal invasive surgery using an endoscope positioning robot. *Methods of information in medicine*, 43(4):403–408, 2004.

[39] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.

[40] J. Zienkiewicz, A. Davison, and S. Leutenegger. Real-time height map fusion using differentiable rendering. In *IEEE International Conference on Intelligent Robots and Systems*, volume 2016-November, pages 4280–4287, 2016.

[41] J. Zienkiewicz, A. Tsiotsios, A. Davison, and S. Leutenegger. Monocular, real-time surface reconstruction using dynamic level of detail. In *Proceedings - 2016 4th International Conference on 3D Vision, 3DV 2016*, pages 37–46, 2016.

[42] S. Zinger, L. Do, and P. H. N. De With. Free-viewpoint depth image based rendering. *Journal of Visual Communication and Image Representation*, 21(5-6):533–541, 2010.

Author Biography

Christian Fuchs received a Diploma degree in Computer Science from the University of Koblenz-Landau in 2011. He works as a research associate in the Active Vision Group. His primary research interests are 3D pose estimation, stereo vision and driver assistance systems.

Dietrich Paulus obtained a Bachelor degree in Computer Science from University of Western Ontario, London, Canada, followed by a diploma (Dipl.-Inf.) in Computer Science and a PhD (Dr.-Ing.) from Friedrich-Alexander University Erlangen-Nuremberg, Germany. He obtained his habilitation in Erlangen in 2001. Since 2001 he is at the Institute for Computational Visualistics at the University Koblenz Landau, Germany where he became a full professor in 2002. His primary interests are computer vision and robot vision.

Free access to this paper is brought to you with the generous support of ON Semiconductor.

All research funding for this paper is referenced in the text; unless noted therein, no research funding was provided by ON.