Perspectively Correct Construction of Virtual Views

Christian Fuchs and Dietrich Paulus

Active Vision Group, University of Koblenz-Landau, Universitätsstraße 1, 56070, Koblenz, Germany
{*fuchsc, paulus*}*@uni-koblenz.de*

Keywords: Virtual Views, Homography Deshadowing, Correct Perspective.

Abstract: The computation of virtual camera views is a common requirement in the development of computer vision appliances. We present a method for the perspectively correct computation of configurable virtual cameras using depth data gained from stereo correspondences. It avoids unnatural warping of *3-D* objects as caused by homography-based approaches. Our method is tested using different stereo datasets.

1 INTRODUCTION

The installation position and orientation of optical sensors is a crucial issue when planning a computer vision application. A lot of different requirements have to be taken into account in order to identify appropriate views of the captured scene for the particular scenarios. Yet, it is not always possible to mount a camera in the desired position, e. g. due to geometric limitations. Therefore, it is not possible to capture the desired view directly. Common applications are in top view systems on vehicles of any kind, for example, which utilize cameras installed on the vehicle to compute virtual bird's views of the surroundings.

Conventional approaches use the images from single cameras for the transformation into a virtual camera view. As single camera setups of course do not provide depth information for the captured scene, the transformation to a virtual view is a difficult issue. Most conventional approaches (Section 2) use a flat world assumption which has several drawbacks in the resulting virtual camera's view. Main drawbacks are the imperfect perspective transformation and the unnatural warping of *3-D* objects.

We address this matter and introduce a method for creating virtual camera views using depth information extracted from stereo images. The goal is to create perspectively correct virtual camera views while avoiding unwanted warping due to violated assumptions. Relevant related work is discussed in section 2. The main drawback of state-of-the-art techniques for virtual camera view computation is depicted in section 3. Sections 4 & 5 introduce the geometric setup and the mathematic techniques used. Test results are shown and concluded in sections 6 & 7.

2 RELATED WORK

Several research topics have to be taken into account when creating virtual camera views. Yet, the following paragraphs can only give a short introduction into the relevant aspects.

2.1 Perspective 2-D Warping

A commonly used approach for the computation of virtual camera views is the utilization of perspective transformations as described by Vincent and Laganiere (Vincent and Laganière, 2001) and Hartley and Zisserman (Hartley and Zisserman, 2003): Given a camera C_1 and a camera C_2. Camera C_1 captures an image which shall be converted to the view described by camera C_2. Assuming C_1 looks at a plane on which all objects on the image are located, the plane in *3-D* space can be described using a minimum of four image points $Q_{C_1} = \left\{ q_i \in \mathbb{R}^2 \,\middle|\, i \in \mathbb{N}^+ \right\}, |Q_{C_1}| \geq 4$ on the image plane. The image plane of C_1 (respectively C_2) is thereby 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 $H_{C_1 \to C_2} \in \mathbb{P}^{2 \times 2}$.

Assuming that the cameras to have fixed lenses and a rigid affine transformation between their poses, matrix $H_{C_1 \to C_2}$ can be considered constant.

2.2 Camera Geometry

In order to describe a camera's imaging properties properly, camera models for the particular camera

types are needed. Tsai (Tsai, 1987) and Tsai and Lenz (Lenz and Tsai, 1988) have published fundamental work on camera calibration techniques especially for the pinhole camera model. Their methods can be used to estimate camera parameters. Based upon their work, Zhang (Zhang, 2000) has released an extended approach for camera calibration. Zhang's algorithm is widely used in computer vision. Of course, cameras following another camera model, need adapted calibration techniques. Geyer and Daniilidis (Geyer and Daniilidis, 2001) published a fundamental discussion of the geometry of catadioptric cameras. Scaramuzza (Scaramuzza, 2008) presents a solution for proper calibration of omnidirectional cameras. This method is suitable for catadioptric and fisheye cameras. The camera parameters estimated using the methods mentioned are prerequisite for the connection of camera images to the *3-D* world.

2.3 Virtual Camera and Bird's View Computation

Various publications address the issue of computing a virtual camera's or a bird's eye view from cameras at different locations. A common approach for the computation of virtual camera views is the utilization of homographies (Section 2.1) for the warping process, which is a reasonable assumption e. g. for vehicles, as the street can be assumed to be a plane. Considerable artifacts are created, e. g. for pedestrians on the street. Liu, Kin and Chen (Liu et al., 2008) use single cameras mounted around a car and transform the images using homography matrices, interpolate and finally stich the images to a top view image. Additionally, they propose a virtual fisheye view as a bird's eye view. Thomas *et al.* (Thomas et al., 2011) stitch top view images on a cost-efficient computation system. Sato *et al.* (Sato et al., 2013) use fish-eye cameras and homographies on spatio-temporal data, whereas Li and Hai (Li and Hai, 2011) focus on the calibration of a multi-view bird's eye view. However, the approaches incorporate heavy distortions for objects which violate the flat world assumption.

Virtual camera views are discussed in the field of image-based rendering also. Shum and Kang (Shum and Kang, 2000) give a survey of different approaches towards view interpolation. Laveau and Faugeras (Laveau and Faugeras, 1994) propose view prediction based on the fundamental matrix using two captured images. Zinger, Do and De With (Zinger et al., 2010) discuss a depth based rendering for *3-D*-TV applications based on disparity maps. Vogt et al. (Vogt et al., 2004) use light-fields to improve image quality in image sequences. However, these publications discuss

the rendering of virtual camera views from camera poses nearby the original camera's views, e. g. light positions shift and/or light rotation.

Up to the best of the author's knowledge, no approach towards perspectively correct virtual cameras with *extensively shifted views* was published, yet.

2.4 Stereo Vision

Using a multiple camera setup, it is possible to compute depth information from images using multiple view geometry. Hartley and Zisserman (Hartley and Zisserman, 2003) summarize the principle behind this approach. The basic idea behind it is to make use of known geometric relations between calibrated cameras and to match the image frames taken at a time. By solving the point correspondence problem, keypoints in corresponding frames can be used together with the camera calibration data in order to estimate the *3-D* position of an object relative to the cameras. The precision of the depth estimates is primarily dependent on the cameras, the baseline of the stereo setup and the matching algorithms.

Concerning the matching of the image content, two major approaches exist: On the one hand, the matching of keypoints, computed using algorithms like *Scale-Invariant Feature Transform* (SIFT) (Lowe, 2004), *Speeded Up Robust Features* (SURF) (Bay et al., 2006) or *Oriented FAST and Rotated BRIEF* (ORB) (Rublee et al., 2011), is a possibility.

In general, image features of any kind can be used as long as an assignment among the image views can be established. An example for a feature based approach towards stereo correspondences was presented by Grimson (Grimson, 1985). Horaud and Skordas (Horaud and Skordas, 1989) group features in order to find correspondences.

Approaches using features resp. keypoints for stereo matching usually result in sparse *3-D* data with high precision matching.

On the other hand, an approach incorporating block matching constitutes the second main category. Various algorithms and improvement have been developed and published so far. Results show dense *3-D* data. The *semiglobal matching (SGM)* approach by Hirschmüller *et al.* (Hirschmüller, 2008) is based on mutual information and uses pixel-wise matching. It shows promising precision properties. The algorithm has become popular and has already been adapted to particular scenarios, for example for in-vehicle applications (Scharwachter et al., 2014; Hermann and Klette, 2013). An optimized version of the SGM was presented by Pantilie and Nedevschi (Pantilie and Nedevschi, 2012). Einecke and Eggert (Einecke and

Eggert, 2015) follow a local correspondence approach in order to significantly reduce execution time while maintaining correspondence quality. A lot of publications concerning stereo reconstruction have already been presented. Therefore, this work does not focus on this issue. Several datasets for the evaluation of stereo reconstruction datasets have been published with an appropriate ground truth.

The *KITTI Stereo Benchmark* (Geiger et al., 2012) provides stereo datasets from road scenes for example. It is commonly for benchmarks for benchmarking stereo correspondence algorithm. Pfeiffer, Gehrig and Schneider (Pfeiffer et al., 2013) published the *Ground Truth Stixel Dataset*, which contains annotated stereo sequence datasets of road scenes.

The *Middlebury Stereo Datasets*[1] provide a collection of stereo scenes. We use stereo images from the newly released 2014 datasets which have been presented by Scharstein *et al.* (Scharstein et al., 2014). The datasets contain high resolution stereo images together with camera calibration files and ground truth disparity maps.

3 HOMOGRAPHY SHADOWING

A commonly used approach for the computation of virtual camera views is the utilization of perspective transformations as described in section 2.1.

As introduced in section 1, the perspective transformation using homographies underlies several drawbacks, because the assumption of a flat world is mostly violated in real world scenarios. When defining a virtual camera V, the corresponding point set has to be computed. A reasonable approach is to use a calibration pattern with known geometry here. This way, the points can be calculated for the desired view. Of course, the resulting homography matrix $H_{C_1 \rightarrow V}$ will rely on the flat world assumption for the defined plane, which is usually confronted with *3-D* objects visible in the image.

The main effect of the violated flat world assumption is an unnaturally warped image which suffers from an effect we denote as the **Homography Shadowing Effect**: The shape of the unnaturally warped object is caused by the violated plane assumption. The homography transformation incorporates the assumption, that all pixels are on the same plane in *3-D* space. The resulting effect on the image projection is an effect similar to a shadow effect that occurs when a point light source is installed at the camera's position. The shape of the warped object matches the shape of

---

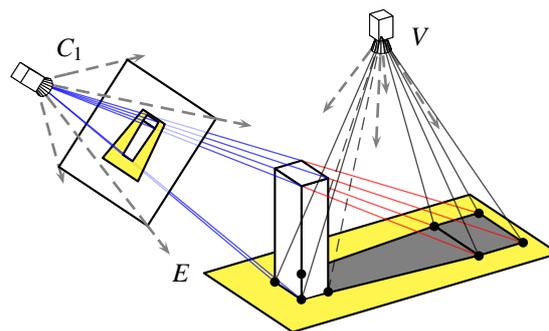[1] http://vision.middlebury.edu/stereo/data/

Figure 1: Homography Shadowing Effect when using the yellow plane as the homography plane.

the resulting shadow that would be caused by the light source at the virtual camera's position.

The effect is illustrated in Figure 1: Camera C_1 captures the image as depicted on its image plane using a pinhole camera model. The yellow plane E is the homography plane for the flat world assumption which is used in the conventional approach. The viewing rays of C_1 (blue) of course end at the object. As no depth information is available, the object will be mapped onto plane E (flat world assumption) when applying the transformation. The view of virtual camera V is computed using a homography transformation. Therefore, V will see the object in shape of its *homography shadow* (gray). An example for this effect is shown in Figure 4.c.

4 VIRTUAL CAMERA GEOMETRY

In order to create virtual camera views, a definition of the virtual camera is needed. The two approaches followed are *perspective* and *orthographic* projection.

For both projections we assume the *z*-axis of the Cartesian coordinate system to be the look direction of the cameras. However, our method can be easily extended to other camera projection models.

The perspective projection model follows the principles of a pinhole camera. It assumes all rays to pass through an infinitesimal hole at the camera's optical center. A camera's geometry and lenses form a field of view

characterized by an intrinsic camera matrix $K \in \mathbb{IP}^{2 \times 2}$ (Hartley and Zisserman, 2003; Lenz and Tsai, 1988; Zhang, 2000; Scaramuzza, 2008):

$$K = \begin{pmatrix} k_x & 0 & c_x \\ 0 & k_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \quad (1)$$

with $(k_x, k_y)^T \in \mathbb{R}^2$ the camera constants in both x and y direction of the pixel grid and $(c_x, c_y)^T \in \mathbb{R}^2$ the camera's principal point. Matrix K is used to describe the projection of *3-D* points to the image plane (*2-D*). In case of a virtual camera, the imaging properties for the cameras have to be defined adequately using a camera configuration space. Figure 2 shows the geometric properties of a pinhole camera. The definition contains five parameters, which are $u, b, l, r, d \in \mathbb{R}$. It is reasonable to assume the camera's view frustum to be symmetric ($l = r$ and $u = b$). This assumption is incorporated in many camera calibration techniques as well (e. g. (Tsai, 1987; Lenz and Tsai, 1988)). The trigonometric relations for the camera opening angles α (horizontal) and β (vertical) are:

$$\alpha = 2 \cdot \mathrm{atan}\left(r \cdot d^{-1}\right) \qquad \beta = 2 \cdot \mathrm{atan}\left(u \cdot d^{-1}\right) \quad (2)$$

In combination with the virtual camera's target image resolution $(w_V, h_V)^T \in \mathbb{N}^2$, the imaging process of the virtual camera can be described. The aspect ratio of a pixel is assumed to be $1 : 1$ (quadratic) in the following paragraphs.

The ratio between w_V and h_V is dependent of the ratio between α and β. Therefore, the configuration space can be narrowed down to:

$$\mathrm{conf}_p(V) = (w_V, h_V, \alpha)^T \quad (3)$$

Using this camera definition, a perspective projection matrix $F_V \in \mathbb{P}^{2 \times 2}$ can be formulated with $F_V = F(\mathrm{conf}_p(V))$ as a diagonal matrix:

$$F_V = \begin{pmatrix} (\tan(0.5\alpha))^{-1} & 0 & 0 \\ 0 & w_V(h_V \tan(0.5\alpha))^{-1} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$
$$(4)$$

This matrix is used for the perspective projection of the *3-D* points into the virtual camera's image. An orthographic projection follows a simple idea of projection. The *3-D* points are transformed onto the image plane using orthogonal projection (no perspective). The principle behind an orthographic projection is shown in Figure 3. In general, four parameters are needed for the definition of the camera Figure 2: $u, b, l, r \in \mathbb{R}$. The same assumptions concerning the symmetric properties as above can be made. Assum-
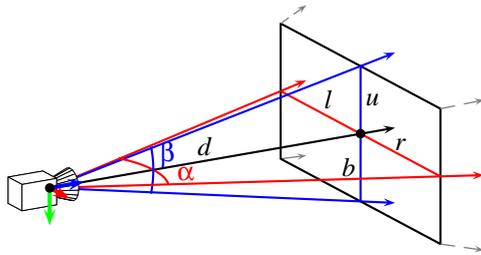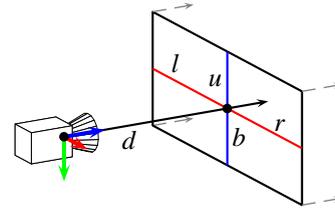

Figure 2: Parametrization of a pinhole camera.

Figure 3: Parametrization of an orthographic camera.

ing a target pixel aspect ratio of 1 (square), the camera is dependent on a scalar factor $s \in \mathbb{R}$. As no perspective parameter is included in an orthographic projection, the configuration is:

$$\mathrm{conf}_o(V) = (w_V, h_V, s)^T \quad (5)$$

The corresponding matrix for orthographic projection $O_V \in \mathbb{P}^{2 \times 2}$ can be defined using $O_V = O(\mathrm{conf}_o(V))$:

$$O_V = \begin{pmatrix} 2(sw_V)^{-1} & 0 & 0 \\ 0 & 2(sh_V)^{-1} & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (6)$$

5 VIRTUAL CAMERA TRANSFORMATION

The position and orientation of an object in a *3-D* space defined by a Cartesian coordinate system can be described using a translation and a rotation relative to the orthonormal bases of the coordinate system. The components can be combined to a so called pose $\mathfrak{a} \in \mathbb{R} \times \mathbb{H}, \mathfrak{a} = (t, \phi)$, with $t \in \mathbb{R}^3, t = (t_x, t_y, t_x)^T$ the translation and $\phi \in \mathbb{H}, \phi = \phi_w + \mathrm{i} \cdot \phi_x + \mathrm{j} \cdot \phi_y + \mathrm{k} \cdot \phi_z$ the rotation as a unit quaternion. The pose definition is used to describe a rigid body transform between two cameras C_1 and C_2.

Given a vector $b \in \mathbb{R}^n$, the vector transformed to homogenous coordinates is represented by $\tilde{b} \in \mathbb{P}^n$. For readability reasons, we use this notation in the following paragraphs and perform some implicit conversions between vectors and their homogenous representation.

Now let C_1 be a calibrated camera with disparities available. These disparities are used for the reprojection to *3-D* space. The goal is to create a perspectively correct view of virtual camera *V*: Given a set of points $P_{C_1} = \left\{ p_{C_1} \,\middle|\, p_{C_1} \in \mathbb{R}^3 \right\}$ in the coordinate system of C_1. The transition to the coordinate system of a virtual camera *V*, whereby pose $\mathfrak{a}$ is the transition from C_1 to *V*, is defined as a function $\Upsilon_{\mathfrak{a}} : \{\mathbb{R}^3\} \to \{\mathbb{R}^3\}$:

$$P_V = \Upsilon_{\mathfrak{a}}(P_{C_1}) \qquad \Upsilon_{\mathfrak{a}}(P) = \{\upsilon_{\mathfrak{a}}(p) \,|\, \forall p \in P\} \quad (7)$$

with $\upsilon_{\mathfrak{a}}(p) : \mathbb{R}^3 \to \mathbb{R}^3$ the transformation of a single vector from C_1 to V. Let $\theta : \mathbb{R}^3 \times \mathbb{H} \to \mathbb{P}^{3\times3}$ compute the transformation matrix according to pose $\mathfrak{a}$. Function $\upsilon_{\mathfrak{a}}(p)$ is defined as:

$$\upsilon_{\mathfrak{a}}(p) = \theta(\mathfrak{a}) \cdot \tilde{p} \qquad (8)$$

The projection onto the image plane of camera V is described by function $\Gamma_V : \{\mathbb{R}^3\} \to \{\mathbb{R}^2\}$:

$$Q_V = \Gamma_V(P_V) \qquad \Gamma_V(P_V) = \{\gamma_V(p) \,|\, \forall p \in P_V\} \qquad (9)$$

with $\gamma_V : \mathbb{R}^3 \to \mathbb{R}^2$ the projection function for camera V for a single point p and $Q_V = \{q_V \,|\, q_V \in \mathbb{R}^2\}$ a set of points on the image plane of V. Let λ_V describe the projection matrix for C, which is dependent on the desired projection:

$$\gamma_V(p) = \lambda_V \cdot p \qquad (10)$$

$$\lambda_V = Z_V \cdot \begin{cases} F_V & \text{perspective projection} \\ O_V & \text{orthographic projection} \end{cases} \qquad (11)$$

with $Z_V \in \mathbb{P}^{2\times2}$ the transformation matrix to pixel coordinates with respect to the image resolution of V:

$$Z_V = \begin{pmatrix} 0.5w_V & 0 & 0.5w_V \\ 0 & 0.5h_V & 0.5h_V \\ 0 & 0 & 1 \end{pmatrix} \qquad (12)$$

In case of a perspective projection, $K_V = Z_V \cdot F_V$ holds. The resulting transform from the *3-D* points P_{C_1} to the view of V is:

$$Q_V = \Gamma_V \circ \Upsilon_{\mathfrak{a}}(P_{C_1}) \qquad (13)$$

The steps above perform the transformation from one camera into the view of a virtual camera. However, it might be desirable to transform multiple source cameras to one virtual view. Our approach can be extended to work with multiple source cameras: Given a set of cameras $\{C_1, ..., C_n\}$ with $n \in \mathbb{N}^+$ and the corresponding set of poses $\{\mathfrak{a}_1, ..., \mathfrak{a}_n\}$, the view of virtual camera V is defined as:

$$Q_V = \Gamma_V \left(\bigcup_{i=1}^{n} \Upsilon_{\mathfrak{a}_i}(P_{C_i}) \right) \qquad (14)$$

Using the transformations described, the view of a virtual camera can be computed out of one or more cameras, when depth/disparity data is available.

6 TEST RESULTS

The virtual camera transformation method proposed was tested using public available datasets from the Middlebury Stereo Datasets

(a) original image (Scharstein et al., 2014)

(b) orthographic virtual view

(c) homography top view (Hartley and Zisserman, 2003)

(d) orthographic virtual top view

Figure 4: Result for the dataset "playtable".

Scharstein *et al.* (Scharstein et al., 2014). We show raw results which were computed using the method we propose. This means that no interpolation for missing parts of the virtual image is performed, as we regard this a post-processing matter which would interfere with the projection quality. In a first test step, the goal is to just alter the projection model of the images. An orthographic projection matrix was configured while the relative pose $\mathfrak{a}$ was set to the zero-pose (identity). Figure 4.b shows an example for the orthographic projection. In the second run, a virtual view is computed. As the datasets unfortunately do not contain the cameras' world positions, pose $\mathfrak{a}$ had to be manually estimated. Nevertheless, the projection properties are shown adequately.

The virtual top views use an orthographic projection and show the transformation to a bird's view perspective. Of course, both perspective and orthogonal projections were investigated. An example result for an orthographic top view is given together with a classically computed homography top view in Figure 4. The configured virtual camera pose is shifted and rotated by $60°$ which of course leads to a sparse pixel density in the areas hidden in the original view. In comparison to the heavily distorted homography top view, we consider the transformation as proposed in this publication to be superior. More results computed using different datasets – including shifted and rotated virtual perspective views – are shown in Figure 5.

7 CONCLUSION

We present a solution for the computation of virtual camera views using stereo data. Using depth data
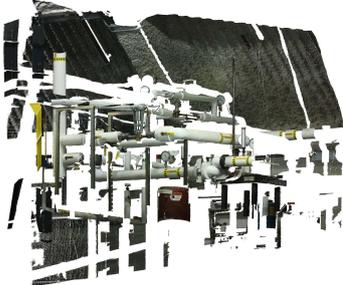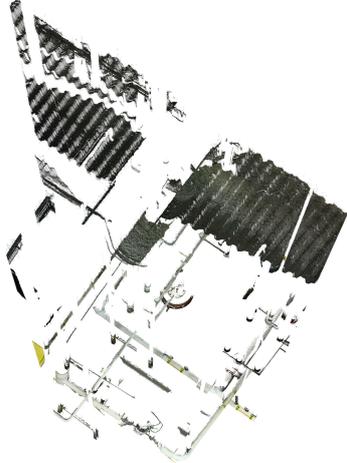
original image (Scharstein et al., 2014)	orthographic projection	virtual camera view
dataset "pipes" – right: orthographic top view		
dataset "recycle" – right: orthographic top view		
dataset "piano" – right: perspective projection from different camera pose		
dataset "shelves" – right: perspective view from different camera pose		

Figure 5: Example transformation results for datasets form the Middlebury Stereo Datasets.

gained from stereo correspondences, it is possible to create perspectively correct images from other camera poses and configurations. The resulting images can be used in various appliances. The density of the pixels in the computed images is dependent of pose α, so that virtual cameras close to the original camera

will lead to (optical) better results. The incorporation of adequate interpolation technologies, such as image inpainting, can improve the quality of the images. As stated in section 6, this is not the current goal of our method, but subject of our current research.

REFERENCES

Bay, H., Tuytelaars, T., and Gool, L. V. (2006). SURF : Speeded Up Robust Features. In *European Conference on Computer Vision*, pages 404–417.

Einecke, N. and Eggert, J. (2015). A Multi-Block-Matching Approach for Stereo. In *Intelligent Vehicles Symposium*, pages 585–592.

Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Computer Vision and Pattern Recognition*, pages 3354–3361.

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

Grimson, W. E. (1985). Computational experiments with a feature based stereo algorithm. *Transactions on Pattern Analysis and Machine Intelligence*, 7(1):17–34.

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

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

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

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

Laveau, S. and Faugeras, O. (1994). 3-D scene representation as a collection of images. *International Conference on Pattern Recognition*, 1.

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

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

Liu, Y. C., Lin, K. Y., and Chen, Y. S. (2008). Bird's-eye view vision system for vehicle surrounding monitoring. *LNCS*, 4931:207–218.

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

Pantilie, C. D. and Nedevschi, S. (2012). SORT-SGM: Subpixel Optimized Real-Time Semiglobal Matching for Intelligent Vehicles. *Transactions on Vehicular Technology*, 61(3):1032–1042.

Pfeiffer, D., Gehrig, S., and Schneider, N. (2013). Exploiting the power of stereo confidences. *Computer Vision and Pattern Recognition*, pages 297–304.

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

Sato, T., Moro, A., Sugahara, A., Tasaki, T., Yamashita, A., and Asama, H. (2013). Spatio-temporal bird's-eye view images using multiple fish-eye cameras. *International Symposium on System Integration*, pages 753–758.

Scaramuzza, D. (2008). *Ominidirectional vision: from calibration to robot estimation*. PhD thesis, Citeseer.

Scharstein, D., Hirschmüller, H., Kitajima, Y., Krathwohl, G., Nešić, N., Wang, X., and Westling, P. (2014). High-Resolution Stereo Datasets with Subpixel-Accurate Ground Truth. *German Conference on Pattern Recognition*, 1(2):31–42.

Scharwachter, T., Schuler, M., and Franke, U. (2014). Visual guard rail detection for advanced highway assistance systems. *Intelligent Vehicles Symposium*, pages 900–905.

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

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

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

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

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

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

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