6D Vision Goes Fisheye for Intersection Assistance

Stefan Gehrig, Clemens Rabe
Daimler AG
Group Research and Advanced Engineering
HPC 050 G 024
71059 Sindelfingen, Germany
{Stefan.Gehrig,Clemens.Rabe}@Daimler.Com

Lars Krüger
Daimler AG
Group Research and Advanced Engineering
Wilhelm-Runge-Str. 11
89081 Ulm, Germany
Lars.Krueger@Daimler.Com

Abstract

For automotive applications, 3D perception of the car's surroundings is crucial, both for driver assistance and for safety systems. In addition, a large field of view is required for future applications such as intersection assistance.

A popular option to obtain 3D measurements and detect objects is to use two cameras (stereo vision). Applications based on object detection range from Adaptive Cruise Control, Collision Mitigation Systems to Intersection Assistance Systems. Stereo vision with conventional cameras only delivers a limited field of view. We extend the field of view using fisheye lenses.

In this contribution a detailed description of a fisheye stereo system for object detection is given, including the steps calibration, rectification, and stereo computation. Differences to traditional stereo systems are pointed out and the accuracy for an experimental setup is evaluated. In addition, the fisheye stereo approach is combined with optical flow measurements to obtain position and velocity data simultaneously. This so-called 6D Vision approach is extended to stereo data obtained with fisheye lenses. Results for intersection scenes are presented.

1. Introduction

Modern vehicles need to be aware of their environment. 3D measurements around the vehicle are therefore needed, including complex traffic scenarios such as intersections. A look at the accident statistics in most countries shows that more than one quarter of all accidents occurs at intersections (see e.g. [3]). Perception tasks for vehicles range from traffic light recognition, traffic sign recognition, lane recognition to traffic participant detection and recognition. We focus on the object detection task.

Solving above tasks, many automotive applications can be implemented such as Adaptive Cruise Control, Stop and Go Driving, Speed Limit Assistance, Collision Mitigation and Intersection Assistance.

3D perception for vehicles is often performed by active sensors such as RADAR and LIDAR which obtain 3D measurements by time-of-flight measurements. Both options can obtain a large field of view but suffer from various problems (see e.g. [7]).

In this paper, we obtain 3D measurements of the surroundings with the use of two passive cameras. In such a case, depth is extracted via triangulation. The cameras are mounted in the wiped area of the windscreen, around the rearview mirror. We use cameras where the exact relative position w.r.t. each other is obtained offline in a software calibration step. Computation time for the calibration step is not relevant, however, the stereo computation of the image pairs must be performed in real-time.

Object detection at unsignalized crossings requires a large field of view in order to detect crossing objects early on (see e.g. [8] for initial approaches). A recent trend to low-cost fisheye sensors makes them interesting for automotive applications.

1.1 Measurement Requirements

For object detection in urban environments, small-sized objects are the most challenging to detect. Taking into account the lower speeds in such environments, these objects have to be detected at 30m distance, which corresponds to the braking distance from 50 km/h to 0 with 0.4g deceleration including some reaction time. Counter traffic has to be detected at larger distances but also has a rather large size. The frame rate of such systems has to be beyond 10Hz for sufficiently fast responses.

For intersection assistance, we determine the field-of-view of interest as follows: While forward moving, we need to monitor the area in front of our vehicle, preferably 180° horizontally. With the assumption, that we only need to respond to threats at speeds beyond 15 km/h, and that laterally intruding vehicles have a speed of 50 km/h maximum, we set for a horizontal field of view of $2 \cdot tan(50/15) \approx 150°$. Vertically, we need a field of view of only 25°, which is roughly determined by the slope variations of the road and the geometry of the camera setting w.r.t. the car body (e.g. the hood). So we need to monitor 25° by 150° in front of the vehicle (see Figure 1 for an illustration). Obviously, the optimal sensor coverage is 360° around the vehicle, but this necessitates a lot of sensors due to self-occlusion.

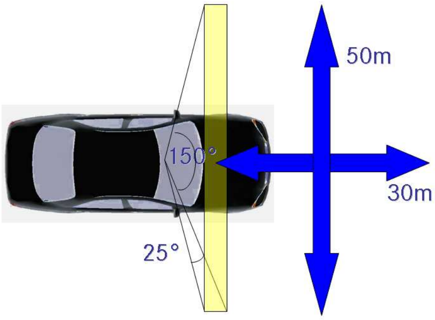

Figure 1. Recommended field of view for advanced driver assistance systems that support the driver at intersections.

1.2 System and Paper Overview

Our proposed system is depicted in Figure 2. For every cycle, a left and right image are acquired (see Figure 3 for an example image pair). We use 1628x400px per image of an 1628x1236 imager. In the next step, the images are rectified (see Section 4), based on the data of calibration performed in an offline process (see Section 3). The rectified images are fed into the stereo engine and a disparity map is obtained which yields a 3D point cloud using the calibration data (see Section 5). Based on that data, object detection can be performed.

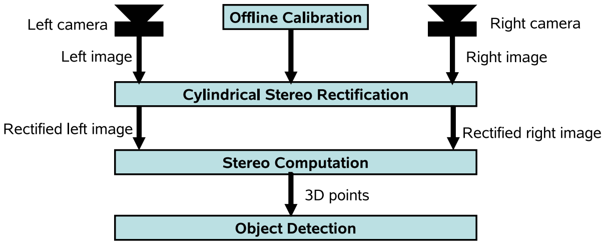

Figure 2. System overview for fisheye stereo computation.

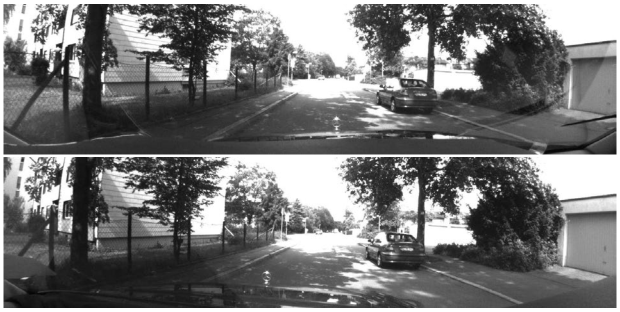

Figure 3. Traffic scene recorded with a stereo fisheye rig (left image on top, right image on the bottom).

We present an extension to this detection scheme in Section 6. This 6D Vision approach fuses optical flow and stereo measurements for fast and precise detection of moving objects. Accuracy measurements of the fisheye stereo system and results of the 6D Vision approach at intersections are shown in Section 7. Conclusions and future work comprise the final section. Before going into details of our system we begin with a literature review in the following section.

2. Related Work

In this brief literature overview we limit ourselves to contributions using stereo on a large field of view. Related work with significant similarity to our approach in calibration or rectification is addressed in the respective sections.

In [7] several options to obtain a large field of view with cameras are investigated. The stereo fisheye is the recommended setup for vehicle applications.

A stereo system for planetary rover navigation is described in [14]. The focus is on efficient implementation of planar rectification via table lookup and stereo. The actual obstacle detection is done via ground plane determination and connected components analysis of protruding obstacles. The implementation obtains a framerate of 3.2Hz.

Fisheye stereo measurements with a planar rectification and stereo combination is demonstrated in [18]. Stereo is computed via template matching. The obstacle detection step is performed via ellipse fitting of the obtained 3D point cloud. No computation time information is provided.

Matuszyk *et al.* [17] use two omnidirectional cameras to perform obstacle detection. Careful adjustment of the sensor is necessary since no software calibration is performed. Only a small portion of the image sensor is used.

In [11] a fisheye stereo rig is presented to investigate confined spaces, such as tubes or fuel tanks. The device is

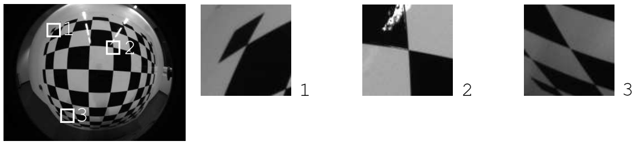

Figure 4. Calibration image and close-ups of chequerboard corners.

only used to display stereoscopic images, hence the alignment requirements are not as strict as for 3D measurements. For the planar rectification used there, the manufacturer‘s distortion curves are utilized, so no software calibration is needed.

3. Calibration

As indicated in the system overview (Fig. 2) the calibration data of the cameras are necessary to perform the rectification and to facilitate a metrically correct scene reconstruction. The calibration data is obtained in an offline step and remains unchanged as long as the camera setup is not changed.

Three issues are of interest with regard to camera calibration: The camera model, the calibration method, and the calibration rig.

The calibration rig and the corresponding rig detection algorithm used here is described by Krüger *et al.* [13]. The approach was originally designed for conventional cameras with strong distortions, but yields also very good results when applied to images of fisheye cameras. The rig consists of 9 × 9 black and white fields of 80 × 80 mm size each. It is manufactured from a single sheet of white painted metal with precision cut black decals.

Due to the tremendous distortions in the calibration images (Fig. 4) we added an extra step to obtain the chequerboard corners with better sub-pixel accuracy than [13]. This step performs a non-linear least mean squares fit of a physically motivated corner model to the image content. The method by Lucchese and Mitra [15] can be used alternatively. Refer to Krüger and Wöhler [12] for details of the corner model and an accuracy analysis for this method compared to [13] and [15].

As the used lens is specified by the manufacturer as an equidistant fisheye lens, its mapping function is:

$$r = f\theta\,. \tag{1}$$

with f being the focal length and r the distance from the principal point. Using Eq. 1 the projection functions $\vec{x}_i = \mathcal{P}(\vec{X}_i, \kappa_c)$ of both cameras are formed, where $\vec{x}$ denotes the projected coordinates (in pixels) of the spatial point $\vec{X}$ using the camera parameters κ_c. Please note that κ_1 — the parameters of the right camera — also include the extrinsic calibration parameters, thus the position and orientation of the right camera w.r.t. the left camera.

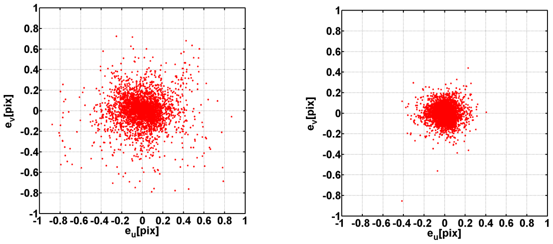

Figure 5. Left: Error distribution without distortion modeling. RMSE=0.171 pixel, MAX=1.10 pixel, Right: Error distribution with distortion modeling. RMSE=0.085 pixel, MAX=0.85 pixel (right)

As the images are to be rectified later on by a table driven implementation the computational complexity of $\mathcal{P}$ does not influence the run time of the rectification. We therefore include the distortion model of Heikkilä and Silven [9] in $\mathcal{P}$ which compensates radial and tangential distortions.

The calibration rig detector yields corresponding points $\tilde{\vec{x}}_i$ and $\vec{X}_i$ for a set of images (about 25 pairs). The calibration performs a non-linear minimization of the least mean square error

$$e = \sqrt{\sum_i \left(\tilde{\vec{x}}_i - \mathcal{P}\left(\vec{X}_i, \kappa_{c_i}\right)\right)^2} \tag{2}$$

using the Levenberg-Marquardt algorithm based on the implementation of Madsen *et al.* [16]. The index c_i denotes the camera the point i was detected in. Derivatives are computed using finite differences. The initial value for the extrinsic parameters rotation and translation is the identity transform. Initial values for the focal lengths are obtained from the pixel size of the camera and the nominal focal length of the lens. Due to the high distortion of the fisheye lenses a large radius of convergence is achieved which does not require a more complicated initialization procedure compared to less distorting lenses.

Fig. 5 depicts the point-wise reprojection error for the calibration used in this contribution. Both variants of $\mathcal{P}$ with and without the distortion term are depicted. The value of e — the root mean square of the point-wise errors — is reduced to about 50% due to the inclusion of the distortion correction.

Please note that this process resembles the well-known Bundle Adjustment method [22] with one difference: The reprojection error is measured in pixel, whereas Bundle Adjustment measures the error in the image plane in e.g. millimeters. However, as the cameras have square pixels and all pixels are of identical size, the error in Eq. 2 is proportional to the error of Bundle Adjustment.

4. Rectification

A planar rectification is the typical choice as preprocessing for stereo computation with approximately parallel orientation resulting in a pinhole image. [6]. For fields of view close to 180° the resulting image size becomes unreasonably large and distorted towards the edge. One example with 150° is shown in Figure 6.

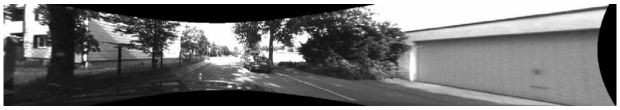

Figure 6. Planar rectification of the left image in Figure 3. Note the shrinking of the parked vehicle in the center and the size increase of the garage on the right.

A straightforward way to circumvent this problem is to use cylindrical rectification instead of pin-hole rectification, first introduced by Roy *et al.* in [19]. Cylindrical rectification maps the incoming rays on a cylinder instead of a plane. For more details on cylindrical rectification and the correct pixel-sampling strategy refer to [2]. See Figure 7 for a top view of such a setup.

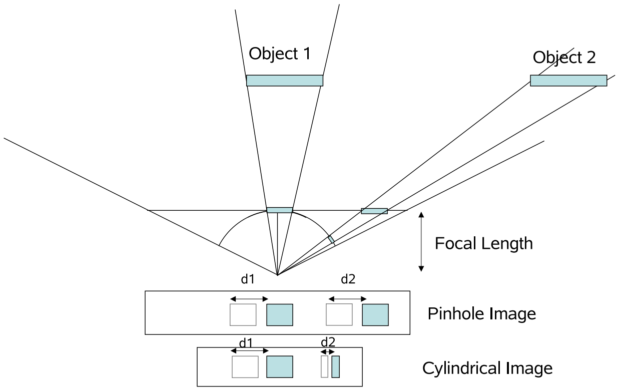

Figure 7. Cylindrical projection principle as used in cylindrical rectification

All that needs to be done is to remap the large pinhole image (which is not needed to be generated at all) to the cylindrical image (shown on the bottom of Figure 7). In addition to planar rectification, only the horizontal image coordinate u has to be mapped according to

$$u_{\text{pinhole}} = f \cdot \tan\left(\frac{u_{\text{cylindrical}}}{f}\right)), \tag{3}$$

with u_{model} being the horizontal image coordinate in the respective camera model, f being the focal length of the fisheye lens according to the equidistant fisheye camera model, setting the principal point to be $(0, 0)$ for simplicity. For small u, pinhole and cylindrical image coordinates differ little whereas for coordinates larger than the focal length of the fisheye lens the changes become drastic (see Object 2 in Figure 7). For the vertical image coordinates no change occurs. The resulting image obeys the formation law of a cylindrical camera model.

Figure 8 shows a cylindrical rectification of the left image from Figure 3. The image looks only slightly modified in size and distortion compared to the original fisheye image, so image information is better preserved than for pinhole rectification.

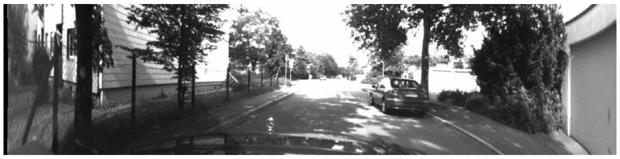

Figure 8. Cylindrical rectification of the left image in Figure 3. The scene appears similar to the original image but epipolar geometry along lines is obtained.

The epipolar geometry of corresponding points in the same image line is maintained in that setup, so the subsequent stereo matching algorithm is not affected significantly by the change in rectification.

5. Stereo Computation

For stereo computation, we investigate both a local and a global stereo algorithm.

5.1 Local Stereo Algorithms

As an example for a local stereo method we employ a hierarchical correlation scheme as described in [4] with a mean-free sum of squared differences. The mean-free part is important due to reflections from the windshield that yield different intensities in the two images.

Local correlation schemes have no problem with cylindrical images. Since the matches are still to be found in the same line, the correspondence search is not altered. However, matches for points that are near the stereo rig resulting in very large disparities cannot be established. This is due to the fact that such points are differently distorted in one image than in the other. In practice, this limitation is of little interest. Even post-processing steps enforcing ordering or uniqueness constraints are barely affected by the fact that images are not compatible with the pinhole camera model.

One thing to consider is the maximum disparity - fewer disparities are needed towards the edge of the image (see disparity d2 in Figure 7 compared to d1 in the same figure). We assume a constant noise in disparity levels (about 0.5) due to sub-pixel inaccuracies. This is correctly reflected in the 3D reconstruction using cylindrical rectification. With planar rectification, these uncertainties would be too optimistic towards the edges of the rectified image, since the resulting large disparities (d2 in Figure 7) are based on image values that are interpolated (see the garage in Figure 6). For an example disparity map corresponding to Figure 8 see Figure 9. After having obtained disparities, 3D points can easily be computed using triangulation.

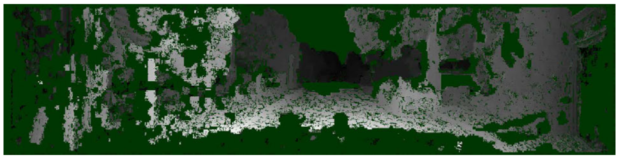

Figure 9. Local stereo result - unmatched areas are marked in dark green, the closer the object the brighter.

5.2 Global Stereo Algorithms

Global stereo algorithms assume a piecewise constant depth for the 3D scene and penalize depth discontinuities in their optimization. Problems arise with cylindrical stereo images as input when this so-called smoothness constraint is applied on disparity level. A wall at a constant distance to the camera system yields decreasing disparities towards the outer edges of the cylindrical image. The change of disparity at constant depth is not accounted for in globally optimal stereo algorithms such as belief propagation of graph cut. Still, since the disparity change vs. image column is smooth, a good reconstruction can be obtained. One example result is shown in Figure 10 corresponding to the image of Figure 8. We use the semi-global matching (SGM) algorithm with Mutual Information as similarity measure to generate the disparity map [10]. The street surface, the car and the lateral objects are correctly reconstructed.

SGM performs an energy minimization in a dynamic-programming fashion on multiple (8 or 16) 1D paths approximating the 2D image. The energy consists of three parts: a data term for photo-consistency, a small smoothness energy term for slanted surfaces that change the disparity slightly (parameter P_1), and a smoothness energy term for depth discontinuities (parameter P_2). SGM takes a few seconds to compute for VGA image pairs on standard CPUs.

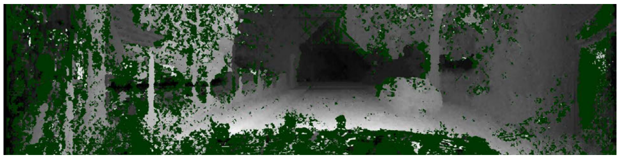

Figure 10. Global stereo result - unmatched areas are marked in dark green, the closer the object the brighter.

6. 6D Vision

So far, only 3D information is obtained using stereo. Based on this data and knowing the position and orientation of the cameras in the vehicle, object detection can be performed (for possible algorithms see [4], [14], or [18]). However, it is very hard to separate two objects close together in depth into one object in motion and one object at rest (see [5] for a good example). Hence we also exploit optical flow to perform object detection to resolve such situations using the 6D Vision concept.

In [5] the 6D vision concept was introduced. We review the approach for a pinhole camera model first. The extension to a cylindrical camera model is shown in the following.

6.1 6D Vision Overview

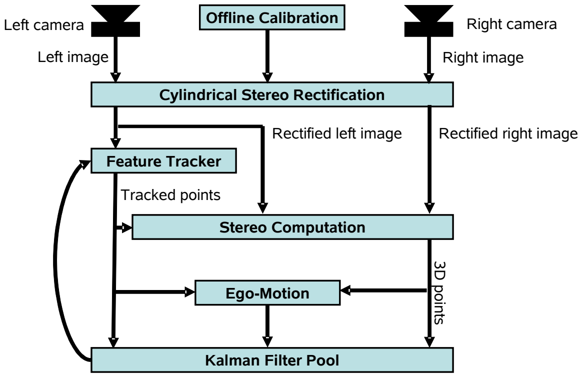

Figure 11. The 6D-Vision system.

The block diagram in Figure 11 shows the main components of the proposed system. Each cycle a new stereo image pair is obtained and the left image is first analyzed by the tracking component. It identifies image features and tracks them over time. In the current application we use a version of the Kanade-Lucas-Tomasi tracker [21] which provides sub-pixel accuracy and tracks the features robustly for a long sequence of images. It was optimized for speed allowing the complete system to analyze up to 5000 features in real-time. The disparities are determined for all features in the stereo module. Here the hierarchical correlation-based from [4] is used. SGM yields better results but is not used here due to real-time restrictions. After this step the current 3D-position of each analyzed feature is known.

In combination with the set of 3D-positions of the last frame these features are used to compute the observers ego-motion. One option to accomplish this is to match the clouds of static world points using optimal rotation and translation estimation [1]. The movement needed to match these point clouds corresponds to the observed camera motion. Instead of using an image-based ego-motion calculation it can be reconstructed using inertial sensor data only. However, using only the car's inertial sensors results in a less accurate estimation of the 3D motion.

The measurements of the tracking and the stereo module together with the calculated ego-motion are given to a Kalman filter system. For each feature one Kalman filter estimates the 6D state vector consisting of the 3D-position and the 3D-motion vector. A detailed description of the underlying models is given in [5] and is recapitulated briefly in the following section. In addition, the covariance matrix for each state vector is available representing the uncertainty of the estimation. This information is important for further processing steps to build probabilistic models of the perceived world.

For the next image pair analysis, the already acquired 6D information is used to predict the image position of the features in the tracker. This yields a better tracking performance with respect to speed and robustness.

Features get lost over time as they move out of the image or become occluded by other image portions. To replace these, the feature detector searches each image for features that are good to track. In our case a gradient based tracker is used and therefore the eigenvalues of the gradient matrix are evaluated according to [20]. As we want to concentrate mainly on moving objects and determine their motion quickly and accurately, it is preferable to have as much information as possible about these objects. Therefore the feature detector increases the density of features in image areas known to have object motion.

6.2 Fusion of Optical Flow and Stereo

In the following we use a right-handed coordinate system with the origin on the road. The lateral x-axis points to the left, the height axis y points upwards and the z-axis represents the distance of a point straight ahead. This coordinate system is fixed to the car, so that all estimated positions are given in the coordinate system of the moving observer. The camera is at $(x, y, z)^T = (0, height, 0)^T$ looking in the positive z-direction.

6.2.1 System Model

Let $\vec{p}_k = (X, Y, Z)^T$ be the 3D position of an observed world point and $\vec{v}_k = (\dot{X}, \dot{Y}, \dot{Z})^T$ its associated velocity vector at the time step k. Combining the location $\vec{p}_k$ and the velocity $\vec{v}_k$ in the 6D state vector $\vec{x}_k = (X, Y, Z, \dot{X}, \dot{Y}, \dot{Z})^T$ we can setup a time discrete linear system model for 3D points [5]. It assumes a constant velocity for the tracked points and an ego-motion module to describe the motion w.r.t. a reference frame at rest.

6.2.2 The Measurement Model

The measurements consist of two pieces of information: the image coordinates u and v of a tracked feature and the disparity d delivered by stereo vision working on rectified images. Assuming a pin-hole camera the non-linear measurement equation for a point given in the camera coordinate system is

$$\vec{z} = \begin{bmatrix} u \\ v \\ d \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} Xf_u \\ Yf_v \\ bf_u \end{bmatrix} + \vec{\nu} \tag{4}$$

with the focal lengths f_u and f_v in pixel and the baseline b of the stereo camera system. The noise term $\vec{\nu}$ is assumed to be Gaussian white noise with covariance matrix R.

To improve the Kalman filters' rate of convergence a multi filter system is used. It consists of multiple differently parameterized Kalman filters running in parallel. Each Kalman filter corresponds to a different object motion model, .e.g. to no motion or a constant motion. By analyzing the innovation of each filter the best matching estimation is chosen. A detailed description of this approach is given in [5].

6.3 Fisheye 6D Vision

The only change of above mentioned mechanism occurs in the measurement model where 3D positions are obtained from disparity measurements. Equation 4 changes to

$$\begin{bmatrix} u \\ v \\ d \end{bmatrix} = \begin{bmatrix} f\arctan(\frac{X}{Z}) \\ f\frac{Y}{Z} \\ f \cdot (\arctan\left(\frac{X}{Z}\right) - \arctan\left(\frac{X-b}{Z}\right)) \end{bmatrix} \tag{5}$$

with $f_u = f_v = f$ for fisheye lenses. Approximating $\arctan(x) = x$ yields Equation 4 again. Now the depth

	laser[m]	stereo[m]	meas.disp.	disp. diff.
grid ur	12.18	11.91	15.7	0.3
grid ll	12.10	11.79	15.8	0.4
grid lr	12.25	11.95	15.7	0.3
l cabinet	6.74	6,76	28.4	-0.1
r cabinet	5.37	5.25	36.3	0.7
plant	4.99	4.87	37.8	0.8
drawer	10.36	9.80	19.0	0.9
r screen	9.70	9.61	19.7	0.2

Table 1. Laser positions vs. stereo measurements (u=upper, l=lower,left, r=right). All disparity errors are within 1px.

does not only depend on disparity but also on the horizontal position. A simplification of the situation can be obtained if virtual pinhole disparities are computed, i.e. the disparities are converted according to the cylinder to pinhole transformation. After that step the original Equation 4 can be used. However, the depth uncertainties for points towards the outer part of the image will be underestimated. For the results presented here, we use this simplification. The correct uncertainties could be incorporated in an adaptive measurement noise matrix.

7. Results

7.1 Stereo Fisheye Accuracy

We compare the absolute stereo accuracy (calibration and disparity determination) with a laser pointer in an indoor scenario. The left image of the scene is shown in Figure 12. The resulting measurements all agree well within a 1px disparity range (see Table 1). An additional source of measurement error in that investigation was the selection of the same point in the image and with the laser pointer in the scene. We expect the errors to be within 0.5px disparity when the reprojection error of the calibration is sufficiently small. This result is similar to our evaluations with a pinhole stereo system [8], taking into account the different focal lengths of the system.

The comparison against ground truth indoor has been performed with a baseline of 30cm. Here, 0.5px disparity uncertainty corresponds to 2.5m uncertainty at 30m distance for the center image part. For the outdoor scenario we chose a baseline of 50cm to cover the necessary measurement volume with more precision (1.7m uncertainty at 30m distance, i.e.less than 6% relative error). With $disparity = \frac{focal_length \cdot baseline}{distance}$ one finds the product of baseline and focal length being the key factor for accuracy. We obtain a value of 250 m·px, similar to [8].

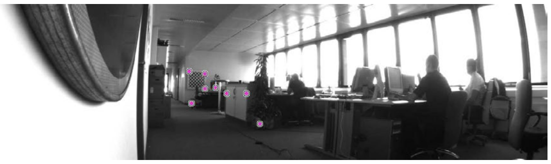

Figure 12. Lab scene (left image) used for laser comparisons with measure points marked as dots in purple.

7.2 6D Vision Results

Figure 13 shows an intersection scene where a vehicle approaches quickly from the right having the right-of-way. Note the position of the vehicle at initial detection. It is first detected at 15m longitudinal and 22m lateral distance, yielding 26m Euclidian distance. An earlier detection was impossible due to occlusion of a wall visible at the right edge of Figure 13.

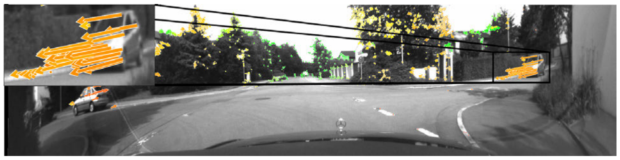

Figure 13. 6D Vision result for a scene with a vehicle approaching fast from the right side having the right-of-way. The significant lateral motion is detected within 4 frames. Arrow length shows the predicted position in 0.5s. The arrow color encodes distance - red is near and green is far away.

The actual object detection based on this very rich information is done via direction and position analysis of the 6D vectors with simple clustering. Five 6D vectors close together in space and direction suffice to robustly detect the object without establishing phantom obstacles (false positives) due to flow or stereo mismatches. Figure 14 shows the same scene two seconds later. The ego-vehicle came almost to a stop and the vehicle from the right is able to continue on its way.

Our second scenario covers an intersection with crossing pedestrians (see Figure 15). All moving objects are detected.

We used up to 5000 features in these scenarios. Tracking needs about 10ms whereas stereo computation takes about 30ms. Another 10ms are needed for rectification and 20ms for the Kalman filtering resulting in a frame rate well above 10Hz on an off-the-shelf PC.

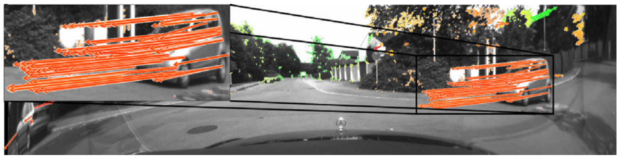

Figure 14. 6D Vision result for the previous intersection scene two seconds later. The ego-vehicle came to a stop, the vehicle from the right side continues on its path unobstructed by the ego-vehicle.

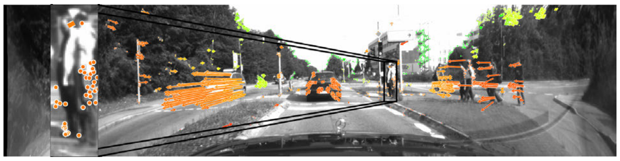

Figure 15. 6D Vision result for an intersection scene with pedestrians approaching from the right. Since the pedestrians close to the driving corridor came to a stop it is safe to continue driving.

8. Conclusions and Future Work

In this paper we presented a fisheye stereo vision system that is able to cover a 150° field of view with less than 6% relative error at 30m distance. The 6D Vision concept has been extended to cylindrical camera models and has shown to be effective for object detection at intersections..

In the future we will use the full measurement equation for the cylindrical camera model to obtain more correct and probabilistically sound 6D measurements at the outer parts of the image.

References

[1] H. Badino, U.Franke, C.Rabe, and S.Gehrig. Stereo-vision based detection of moving objects under strong camera motion. In *International Conference on Computer Vision Theory and Applications*, pages 253–260, Setubal, Portugal, February 2006.

[2] B. Bartczak, K. Koeser, F. Woelk, and R. Koch. Extraction of 3d freeform surfaces as visual landmarks for real-time tracking. *Journal of Real-Time Imaging*, 2(2):81–101, 2007.

[3] Bundesamt für Statistik. *Verkehr in Zahlen 2000 (Datengrundlage 1999)*. Deutscher Verkehrs-Verlag, 2000.

[4] U. Franke. Real-time stereo vision for urban traffic scene understanding. In *IEEE Conference on Intelligent Vehicles*, Dearborn, October 2000.

[5] U. Franke, C. Rabe, H. Badino, and S. Gehrig. 6d vision: Fusion of motion and stereo for robust environment perception. In *Pattern Recognition, DAGM Symposium 2005, Vienna*, pages 216–223, 2005.

[6] A. Fusiello, E. Trucco, and A. Verri. Rectification with unconstrained stereo geometry. In *British Machine Vision Conference*, 1997.

[7] S. K. Gehrig. Large-field-of-view stereo for automotive applications. In *OmniVis 2005, Beijing*, October 2005.

[8] S. K. Gehrig, J. Klappstein, and U. Franke. Active stereo for intersection assistance. In *Vision Modeling and Visualization Conference, Stanford, USA*, pages 29–35, November 2004.

[9] J. Heikkila and O. Silven. A four-step camera calibration procedure with implicit image correction. In *CVPR97*, pages 1106–1112, 1997.

[10] H. Hirschmueller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *Proceedings of Int. Conference on Computer Vision and Pattern Recognition 05, San Diego, CA*, volume 2, pages 807–814, June 2005.

[11] Z. Kiraly, G. Springer, and J. VanDam. Stereoscopic vision system. *Optical Engineering*, 45(4):1–13, April 2006.

[12] L. Krüger and C. Wöhler. Accurate chequerboard corner localisation for camera calibration and scene reconstruction. *Submitted to Computer Vision and Image Understanding, Special Issue: Intelligent Vision System*, 2008.

[13] L. Krüger, C. Wöhler, A. Würz-Wessel, and F. Stein. Infactory calibration of multiocular camera systems. In *SPIE Photonics Europe*, pages pp. 126–137, 2004.

[14] L. Liu, J. Jioa, and L. Li. An obstacle detection system using binocular stereo fisheye lenses for planetary rover navigation. In *36th COSPAR (Committee on Space Research) Scientific Assembly, in Beijing, China*, July 2006.

[15] L. Lucchese and S. Mitra. Using saddle points for subpixel feature detection in camera calibration targets. In *Asia-Pacific Conference on Circuits and Systems*, 2002.

[16] K. Madsen, H. B. Nielsen, and O. Tingleff. Methods for non-linear least squares problems, jul 1999.

[17] L. Matuszyk, A. Zelinsky, L. Nilsson, and M. Rilbe. Stereo panoramic vision for monitoring vehicle blind-spots. In *Proceedings of the Intelligent Vehicles 04 Symposium*, pages 31–36, 2004.

[18] T. Nishimoto and Y. Junichi. Three-dimensional measurement using fisheye stereo vision. In *SICE - Technical Meeting on Systems and Control, IEE Japan*, 2007.

[19] S. Roy, J. Meunier, and I. J. Cox. Cylindrical rectification to minimize epipolar distortion. In *Proceedings of Int. Conference on Computer Vision and Pattern Recognition 97*, pages 393–399, 1997.

[20] J. Shi and C. Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, June 1994.

[21] C. Tomasi and T. Kanade. Shape and motion from image streams: a factorization method – 3. detection and tracking of point features. Technical Report CMU-CS-91-132, School of CS – CMU, April 1991.

[22] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment - a modern synthesis. In *ICCV '99: Proceedings of the International Workshop on Vision Algorithms*, pages 298–372, London, UK, 2000. Springer-Verlag.