Accurate 3D-Vision-Based Obstacle Detection for an Autonomous Train

Johann Weichselbaum[a], Christian Zinner[a], Oliver Gebauer[b], Wolfgang Pree[b]

[a]*Safety & Security Department, AIT Austrian Institute of Technology GmbH, Donau-City-Straße 1, 1220 Vienna, Austria, {johann.weichselbaum, christian.zinner}@ait.ac.at, Tel.: +43 50550 4120, Fax.: +43 50550 4250*
[b]*Department of Computer Science, Univ. of Salzburg, Jakob Haringer Straße 2, 5020 Salzburg, Austria*

Abstract

In this paper we present a 3D-vision based obstacle detection system for an autonomously operating train in open terrain environments. The system produces dense depth data in real-time from a stereo camera system with a baseline of 1.4 m to fulfill accuracy requirements for reliable obstacle detection 80 m ahead. On an existing high speed stereo engine, several modifications have been applied to significantly improve the overall performance of the system. Hierarchical stereo matching and slanted correlation masks increased the quality of the depth data in a way that the obstacle detection rate increased from 89.4% to 97.75% while the false positive detection rate could be kept as low as 0.25%. The evaluation results have been obtained from extensive real-world test data. An additional stereo matching speed-up of factor 2.15 was achieved and the overall latency of obstacle detection is considerably faster than 300 ms.

Keywords: Dense Stereo Matching, Census Transformation, Slanted Correlation Masks, Autonomous Train, Obstacle Detection

1. Introduction

This work has been carried out in the context of a research project that aims to equip trains, especially region branch lines, with complex sensor systems to enable autonomous operating on existing routes without the need of major constructional modifications of the track system. Autonomous trains

as such are not new, however, existing systems usually operate on track systems which are closed hermetically to prevent any persons or animals from getting into the endangered area. Within the proposed concept, the system must be aware of any obstacles which may appear, and suitable decisions and actions have to be carried out on the basis of extensive sensor information.

The major motivation for the development of an autonomously driving train is the low economy of regional trains due to their sparse occupation, which is partially caused by their low frequency. Trains arriving in an interval of 10 minutes instead of one or even two hours could double or triple their passenger demand as they would not require any knowledge about their schedule and be available nearly instantly. These vehicles could be smaller and be operated as powered rail cars. Their economy could be raised significantly if there were not the need for an increased amount of drivers. Replacing the driver's function without additional investments into the rail infrastructure is therefore the key to improved attractiveness of regional trains. Besides of the manipulation of the car the driver's function is the ability to differ between relevant and irrelevant objects on the track and the proper control of the car's speed with regard to the safety of the car and its passengers. Examples for the challenge of judging objects on rails correctly are snow, vegetation between rails, plastic ribbons, blown away newspapers, fallen trees or branches, mudflow, avalanches, other cars, animals or humans. Currently there is no single sensor known which is able to handle this judgment as good as the human driver can do it with the use of his eyes only. On the other hand, technical sensors offer abilities which overstrain the human, as they do not get tired or detracted and can see without the presence of light or even through fog.

1.1. Setup of the Experimental Test Platform

In order to achieve an obstacle recognition performance which comes close to the human's eye a combination of sensors which are based on different technical principles is the best choice. We used laser scanners, single and stereo cameras working in the spectral ranges of visible light and infra-red, radar and ultra sonic sensors (see Fig. 1). Besides the technical characteristics of sensors the effective evaluation and fusion of their signals contributes to a useful recognition performance. In addition to precise 3D sensor data, a high level of precision is also required for the self-localization of the car, as the obstacle recognition depends on an offline-created map of the track's trajectory. For a successful discrimination of objects to be inside or outside the

Figure 1: Test platform

track clearance on curved tracks, the longitudinal position of the car must be known better than 1 m (cf. Fig. 16).

1.2. Sensor Systems for Vision based Obstacle Detection

In this paper, we contribute the development of a highly accurate real-time stereo vision system for obstacle detection as part of the sensor system of the train. In contrast to the other sensor modalities that are available off-the-shelf, the stereo vision sensor itself is a topic under research and it needs application-specific optimization. Therefore, this work is an important precondition for a later evaluation and comparison of all sensors applied on the test platform.

Major challenges for a stereo vision system within this context are

- a short latency time between image capturing and results output
- a high angular resolution to be able to discriminate obstacles from other objects which might be located closely beside the tracks (e.g., poles,

signaling equipment)

- a high amount of depth resolution is necessary to fulfill the previous requirement on curved tracks
- a high inter-frame dynamic range to cope with wide variety of occurring outdoor illumination situations
- a high intra-frame dynamic range to be able to capture high contrasted scenes
- post-processing and analysis of depth data in order to provide the results as a list of objects with annotated attributes according their size and location in 3D.

We propose a solution that employs dense stereo matching techniques while using cameras with a comparably high resolution together with a quite wide stereo baseline.

1.3. Outline

The remainder of this paper is organized as follows. Examples for other stereo vision based sensors for vehicles are quoted in Section 2. In Section 3 basic requirements for this particular application are used to deduce a stereo geometry and camera setup. An initial result uncovers several weaknesses that motivated further improvement. Section 4 describes a number of concrete measures to improve results quality as well as lower the computational effort. Section 5 covers the task of extracting information about obstacles from the depth images resulting from stereo matching. Extensive evaluations from large real-world datasets are reported in Section 6. Concluding statements are given in Section 7.

2. Related Work

Stereo vision based obstacle detection is a popular technology for advanced driver assistance systems. There is a wide variety of possible approaches (feature-based vs. dense stereo, combination with optical flow, variants of SLAM, etc.) that have been discussed in a series of publications.

It is very common that real-time capable dense stereo matching relies on a calibration procedure that ensures the epipolar constraint to reduce the matching problem to a one-dimensional search of the correct disparity per

pixel. It is also common that, due to the underlying principle of triangulation, the depth resolution decays proportionally to the square of object distance. This is an issue in inter-urban traffic in railway applications, where commonly used stereo baselines between 0.2 and 0.4 meters simply cannot deliver sufficient depth resolution to differentiate whether distant obstacles are on track or only nearby. The particular issues caused by large baselines while being forced to meet very restrictive computation time constraints are less reflected in literature.

Obstacle detection with dense stereo is done in [1], where the high computational effort of the stereo matching has been tackled with a hybrid SW-HW solution. With a stereo baseline of 0.32 m and an image width of 512, real-time dense stereo matching is shown to be feasible.

[2] calculates dense v-disparity maps by a semi-global matching approach and performs a Hough-transform based analysis for the obstacle detection. Available computing power as of 2002 seems to be the reason for the quite small image width of 380. These two approaches also make use of lane detection to check whether 3D points are inside a certain clearance.

In [3] a highly parallelized disparity engine in hardware is realized to achieve interactive frame rates on VGA images.

The 3D vision group of AIT contributed a stereo vision based obstacle detection during a joint participation with the team of Auburn University at the DARPA Grand Challenge 2005 [4]. A more recent outcome is the real-time mapping approach based on dense stereo vision according to [5].

3. Requirements and Initial Stereo Vision Concept

Within the application scenario, few major a-priori requirements have been identified for the vision-based obstacle system.

- detection of obstacles inside the tracks clearance volume with at least a size of $0.3 \times 0.3 \times 0.3$ m at a distance from 10 m up to 80 m ahead
- localization accuracy of obstacles relative to the sensor system of less than 1 m in the longitudinal direction and less than 0.5 m in the lateral direction
- horizontal field-of-view of 40° in order to capture any relevant scenes according to the smallest curve radius on the test track within 80 m distance

range boundary	disparity	depth	depth step	subpixel depth step	lateral pixel size
near	230 px	9.96 m	0.043 m	0.014 m	0.006 m
far	28 px	81.8 m	2.92 m	0.97 m	0.05 m

Table 1: Initial stereo vision properties based on 1200 pixel wide input images

- latency time of less than 300 ms from capture time to obstacle report time for the first experimental implementation.

Especially the depth resolution requirement is very hard compared to other applications related to autonomous land vehicles. It is necessary to be able to discriminate real obstacles from ordinary infrastructure elements closely to the tracks clearance volume – even in curves. We have to define a stereo camera geometry that meets the requirements. The basic relation between depth (z) in meters and disparity (d) in pixels is

$$z = \frac{T \cdot f}{d} \ , \tag{1}$$

with the focal length f given in multiples of the sensor pixel pitch and the size of the stereo baseline T in meters. A measure for the resolution in longitudinal direction can be derived from the depth step per (subpixel-) disparity step. Using the absolute value of the first derivative of (1) weighted by a subpixel-refinement factor r_s and expressed for z yields

$$R_{long}(z) = r_s \cdot \frac{z^2}{T \cdot f} \ . \tag{2}$$

Supposing an empirically observed enhancement through subpixel refinement of factor 3 ($r_s = \frac{1}{3}$, the required depth resolution of 1 m at a distance of 80 m is still challenging and results in a stereo baseline of 1.4 m and the need of evaluating 1200 pixels wide images. A sensor of resolution 1600×1200 pixels has been chosen. Its pixel pitch of 5.5 µm leads to a focal length requirement of 12 mm. The summary in Table 1 shows, that the chosen configuration meets these requirements.

The current prototype of the camera system can be seen in Fig. 1 above the front window. The outer monochrome cameras form a baseline of 1.4 m. The middle color camera primarily serves as source for color information and can be also exploited for stereo vision.

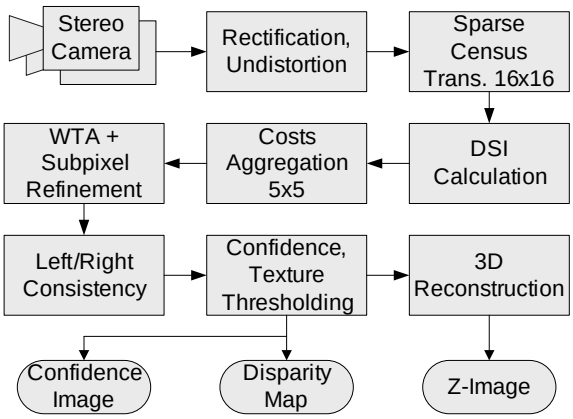


Figure 2: Census based stereo matching block diagram

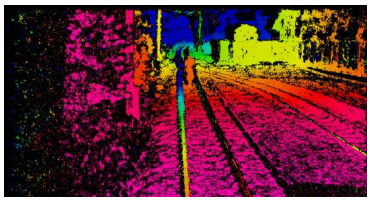
All three cameras have been calibrated in such a way that, after rectification, they fulfill the epipolar constraint, and every scene point is projected onto the same scan line in every camera.

3.1. Correlation Based Stereo Matching

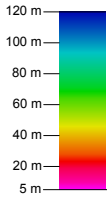
The large image dimensions required in conjunction with hard timing constraints allows only for highly efficient correlation based stereo matching approaches that use local optimization strategies. As a basis of this work, the method described in [6], which is implemented with various performance optimizations as according to [7] in the software stereo engine *S3e*. A good overview of various other existing stereo matching algorithms give [8] and [9].

Fig. 2 shows a simplified block diagram of the stereo matching method that applies a census transform with a relatively large mask size of 15×15. Census-transform based matching has shown robustness according to radiometric differences of the input images, which is a distinct advantage when matching images originating from monochrome cameras against such from bayer-filter-equipped color cameras. The large census mask helps exploiting even low amounts of texture in the scene. The computational effort has been significantly lowered by using a sparse census transform scheme. Stereo matching costs are aggregated by relatively small aggregation windows ranging from 3×3 to 5×5. Matching costs are analyzed according a winner-takes-all strategy which has included a confidence estimation, subpixel refinement and L/R-consistency check facility.

(a) Left rectified image



(b) Depth image



(c) Color legend

Figure 3: Results of reference configuration

3.2. First Test Runs

Fig. 3 shows a camera view of a real-world scene along with a color-coded depth map after the existing standard stereo matching engine has been applied on it. It discloses several problems, which have their main reason in the required depth resolution on long distances:

- Depth data on the ground floor is not very dense, which is mainly caused by perspective distortion between the input images due to the large baseline.
- The ambiguity between the two tracks caused false depth data on the left track and no data on the right one.
- A net computation time of 325 ms per frame on a state-of-the-art PC with an i7 CPU @ 3.06 GHz and 4×2 cores has been achieved. This is too long, although the achieved performance figure of 560 million disparity evaluations per second (1200 × 660 pixels @ 230 disparities @ 3.08 fps) is quite competitive.

Each color in the depth image in Fig. 3 indicates a certain distance. Throughout this paper, the distances in the depth images are coded according to the color bar shown in Fig. 3c.

4. Enhancements for Accurate and Fast Stereo Matching

The aforementioned problems are a matter of results quality as well as computational load. We will identify modifications to the system that deliver improvement for both.

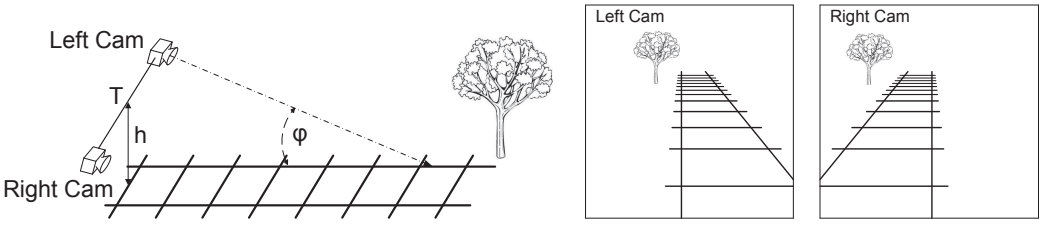


Figure 4: Perspective distortion on ground plane (mounting height h, baseline T, angle between optical axis of the cameras and ground plane φ)

4.1. Slanted Correlation Mask

The large baseline, that is necessary to obtain the required accuracy for this application, leads to extensive perspective distortions on the ground plane. The amount of distortion can be quantified as the slanting-angle α between the projections of the ground plane in the left and right camera images, respectively. According to Fig. 4, α follows the relation

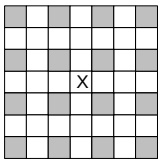
$$\alpha = atan(T \cdot \frac{cos(\varphi)}{h}) \ . \tag{3}$$

Due to the usage of standard quadratic correlation masks, we obtained poor results on the ground plane from the initial *S3e* system. The problem is known in literature, e.g., [10] leverages a simplified Lucas-Kanade approach and also [11] computes a 3D plane at each pixel on which a support region is projected. Our considerations led to the conclusion that such approaches impose too much additional computational cost because any additional variation in the program flow will over-proportionally undo the benefits of our extensive platform-specific optimizations using SSE instructions.

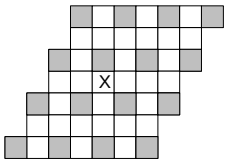
In the case of *S3e* the problem originates much less from the cost aggregation, whose rather small mask sizes limit the effect of disparity gradients within the mask. However, the large census mask sizes are much more suffering from perspective distortion. To overcome this problem we use few discrete variants of slanted sparse census masks (Fig. 5b) and superimpose their results.

$$\alpha = tan(\frac{\Delta u}{\Delta v}) = tan(\frac{1}{2}) \approx 26.56^\circ \ , \tag{4}$$

which is similar to the result of (3) when inserting the actually used stereo camera geometry. The effectiveness is illustrated in Fig. 6, where a scene has



(a) Normal census mask



(b) Slanted census mask

Figure 5: Two different types of sparse census masks

been composed from two planes – one of them parallel to the image plane, and the other one simulating a slanted ground plane. The results of the two census mask variants are finally superimposed on a per-pixel-level according to

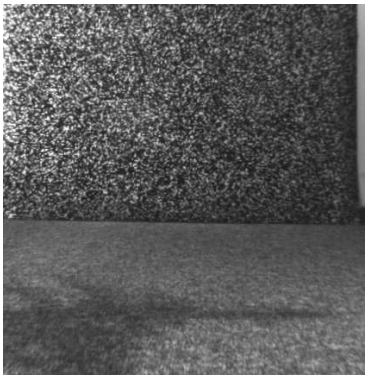
$$d_{result}(u,v) = \begin{cases} d_{normal}(u,v), & c_{normal}(u,v) \geq c_{slanted}(u,v) \\ d_{slanted}(u,v), & else \end{cases} , \qquad (5)$$

where $c(u,v)$ represents the confidence of the disparity value $d(u,v)$.

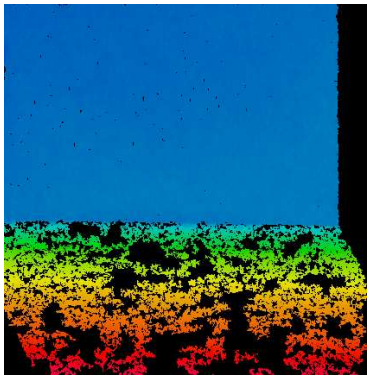
We additionally observed that the matching results are similarly dense and smooth for any orientation of a plane that is in-between the ones in the test scene. Thus, we generalized the approach for other situations occurring that cause major distortions (Fig. 7). The corresponding projections on the left and right image planes are depicted in Fig. 8 and Fig. 9 shows the census masks that correct these distortions during the stereo matching. We chose to treat only one input image with a slanted mask and re-use the intermediate results from the straight masks, which saves some overhead in contrast to using asymmetric masks. The census mask according to Fig. 5b realizes a slanting-angle of Each type of census mask can be individually switched-on in the matching process, which allows to incorporate as less additional computational costs as necessary. Furthermore, when combined with the hierarchical stereo matching approaches of Section 4.3, the overhead can be kept even lower, since the slanted masks can be avoided on the bottom level of the image pyramid when using the *reprojection pyramid mode* as well as when the disparity hint is generated from a significantly smaller stereo baseline according to the dual baseline approach of Section 4.2.

4.2. Dual Baseline Approach

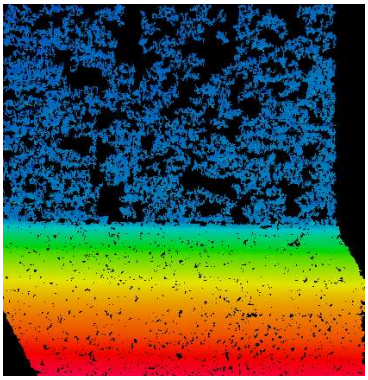
Using a combined approach with a second baseline that is smaller can reduce the computation time for a single frame. The idea is to exploit the three-camera configuration for a smaller distance between the two cameras



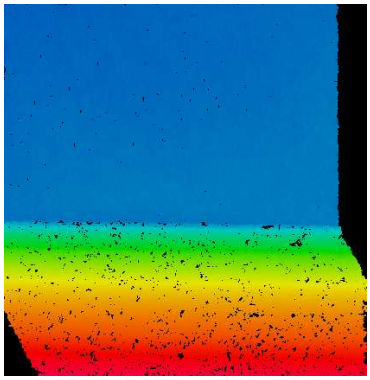
(a) Left rectified image with textured planes



(b) Depth image with normal correlation mask

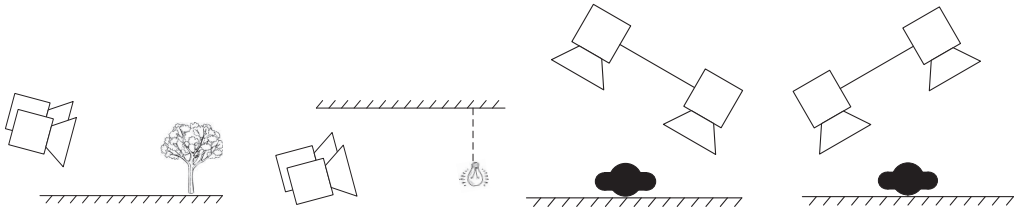


(c) Depth image with slanted correlation mask



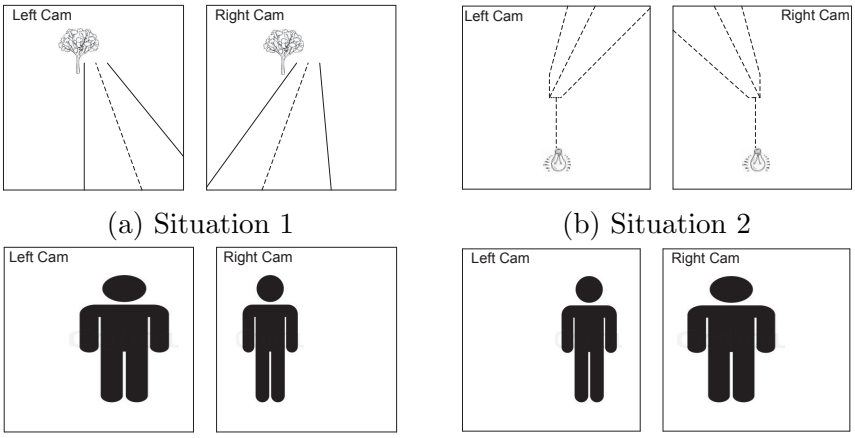
(d) Combination of normal and slanted masks

Figure 6: Depth images using different correlation masks



(a) Situation 1 (b) Situation 2 (c) Situation 3 (d) Situation 4

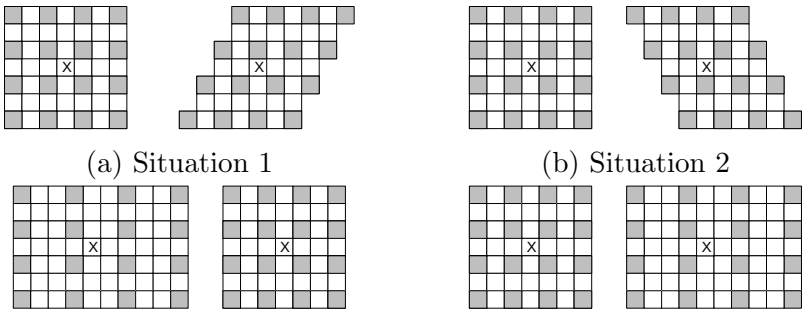
Figure 7: Situations with perspective distortion



(a) Situation 1 (b) Situation 2

(c) Situation 3 (d) Situation 4

Figure 8: Perspective distortions on image planes for different situations



(a) Situation 1 (b) Situation 2

(c) Situation 3 (d) Situation 4

Figure 9: Sparse census masks for differnt types of perspective distortions

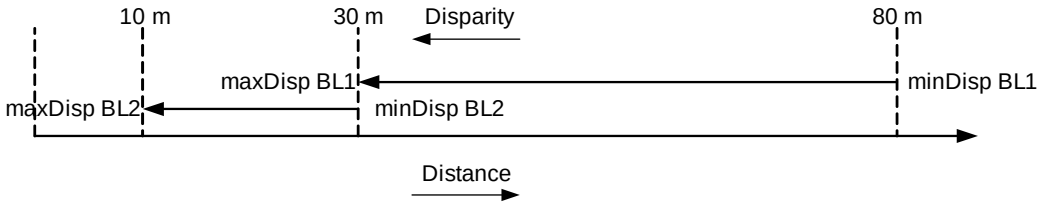


Figure 10: Disparity ranges for both baselines (minDisp BL1/2 and maxDisp BL1/2 are the minimum and maximum disparity values of the searchranges of both baselines

of the second baseline, as this allows for a reduced disparity search range. As the second baseline is intended to observe the near range of the clearance gauge, the accuracy of the system is still guaranteed, even when using a lower image resolution. The disparity range of the large baseline can also be reduced, since it is only responsible for the further distances.

Fig. 10 shows how the whole search range is divided into a near range and a far range interval. Both range intervals define the disparity ranges for the corresponding baselines. The disparity images resulting from each of the baselines are combined to a single final disparity map that is then used for further processing and obstacle detection.

In Fig. 11 the effect of the dual baseline approach can be seen. The first image (Fig. 11a) shows the left rectified input image of the scene. Fig. 11b and Fig. 11c show the resulting disparity images of the large baseline and the smaller baseline. These two images get combined to the final disparity image that can be seen in Fig. 11d, where the valid pixels in the near range and far range area are taken from the disparity image of the small baseline and large baseline, respectively. The decision process is visualized by Fig. 11e, where white pixels correspond to pixels of the larger baseline and grey pixels to disparity values from the small baseline. The final combined depth image of the scene is depicted in Fig. 11f.

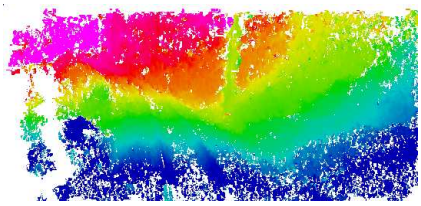
4.3. Hierarchical Stereo Matching

The stereo matching algorithm is over-proportionally computational demanding for high resolution images. As a remedy, a hierarchical stereo matching approach has been implemented, based on the concept described in [5] which in turn was inspired by contributions like [12].

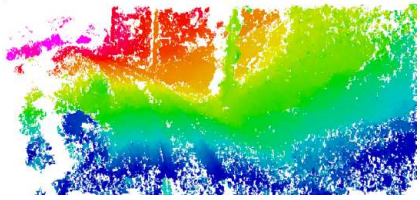
The basic idea can be seen in Fig. 12. Stereo matching is first performed

(a) Left rectified input image

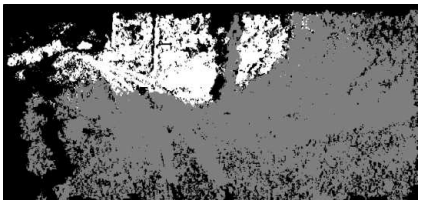
(b) Disparity image baseline 1



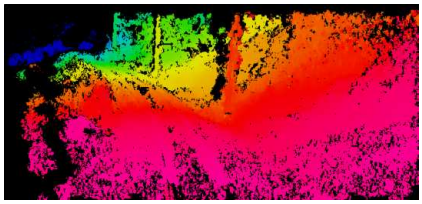
(c) Disparity image baseline 2



(d) Combined disparity image

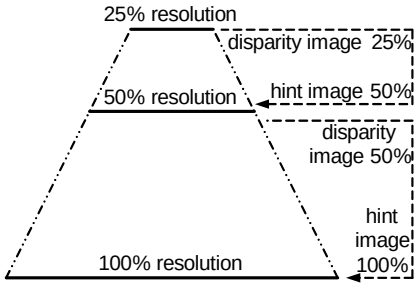


(e) Decision image

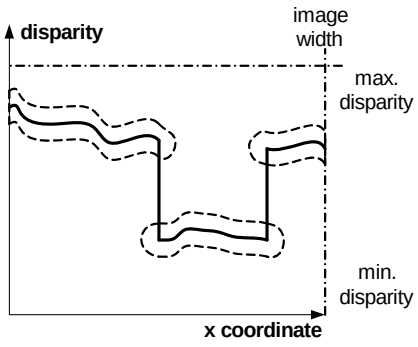


(f) Combined depth image

Figure 11: Example of combined disparity image with dual baseline approach



(a) Hierarchical approach



(b) Reduced disparity range

Figure 12: Hierarchical stereo matching [5]

on down-scaled input images. If, for example the images get reduced by the half in width and height, also the disparity search range is reduced by the same factor. As a consequence, the effort of some steps (e.g. census transform) is reduced to one fourth and some steps (e.g. DSI calculation) even to one eighth at this pyramid level. The resulting disparity image of reduced half resolution is intended to serve as a disparity-hint-image $d_h(x, y)$ for the stereo matching process on the next higher resolution images. According to the information in the disparity-hint-image, only a reduced disparity range (e.g. hint-value $\pm$ 4 disparities) has to be searched for the correct match. However, in order to preserve lateral precision, the search intervals from the hint-image are propagated on a small x-y-neighborhood, which allows a refinement of the lateral position of depth discontinuities, too. Depending on image resolution and the number of hierarchical steps, an overall speed-up-factor of more than 6 has been achieved in [5].

During this work, the approach has been further extended by a so-called *reprojection pyramid mode*. Instead of limiting the disparity search range when matching the original left and right input images $I_l(x, y)$ and $I_r(x, y)$, we now are matching the left input image $I_r(x, y)$ against a re-projected right input image $I_{repr}(x, y)$ using a rather small disparity range of, e.g., ± 4 disparities. The re-projection is a horizontal backward-mapping

$$I_{repr}(x, y) = I_r(x - d_h, y) \ , \tag{6}$$

where the up-scaled disparity hint $d_h(x, y)$ from the prior pyramid level represents the offset in x-direction. Since disparities are subpixel-quantized, the mapping applies linear interpolation. The stereo matching engine is now applied between $I_l(x, y)$ and $I_{repr}(x, y)$, which yields a map of disparity offsets $d_{off}(x, y)$ that can be used to refine the original disparity hint image according to

$$d_{refined} = d_h(x, y) + d_{offs}(x, y) \ . \tag{7}$$

As already indicated, this refinement method works also on disparity hint images with lower resolution. Furthermore, the largest part of disparity gradients due to perspective distortions (as discussed in Section 4.1) are eliminated during the reprojection step, so no slanted correlation masks need to be applied. Thus, the hierarchical- as well as the disparity refinement- as well as the slanted-aspects are combined into a single step.

The only drawback is that the lateral resolution does not become significantly better when compared to the disparity hint image. For this reason,

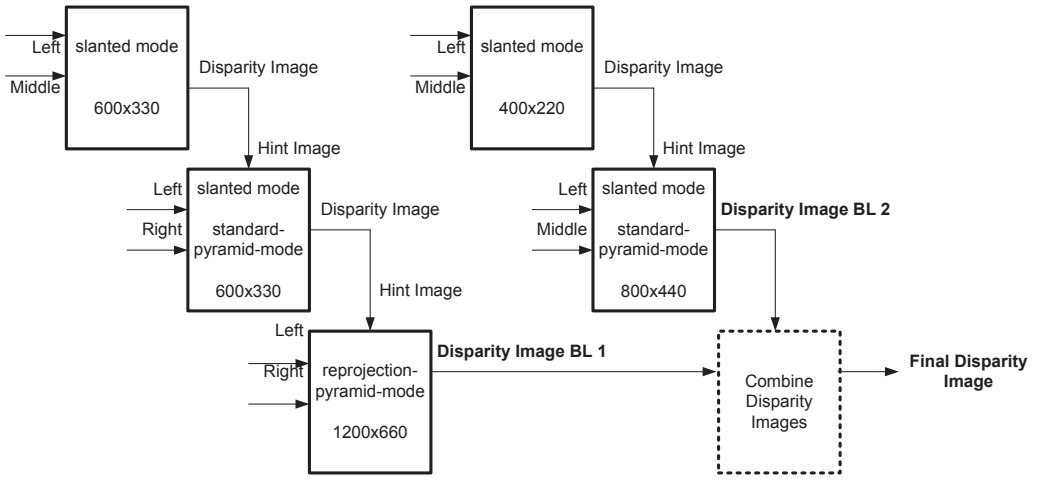


Figure 13: Sequence of operation diagram

the reprojection pyramid mode fits best when used on the highest resolution step of each pyramid matching process.

With this toolbox of possibilities for building a hierarchical processing chain we elaborated a scheme according to Fig. 13, where each box represents a stereo matching process with a certain resolution and mode of operation. Each process uses two input images from either the left and right or left and middle cameras from a certain level of pyramidal resolution decimation and some do also have a disparity hint input image from a different stereo matching process. The left three boxes represent the large baseline (BL1) and the right upper two boxes the small baseline (BL2). BL2 uses one additional hierarchical step with half the resolution of the final disparity map of BL2 of 800 $\times$ 440 pixels which is sufficient to ensure the specified accuracy on the near range. On the contrary, BL1 uses three hierarchical steps with a final output resolution of 1200 $\times$ 660 pixels. For BL1, the first hierarchical step uses the middle camera, while in the remaining steps the right camera is used.

5. Obstacle Detection from Depth Data

5.1. Clearance Filtering

The track on which the train is operating has been recorded in advance as a sequence of GPS points. During operation the train is also equipped with a

GPS receiver for self-localization. Thus, a framework provides the trajectory of the track ahead up to a user-defined distance at each point in time and also for the exact moment the three cameras are synchronously triggered. After image acquisition and disparity image calculation, each valid disparity value is transformed into a 3D-Point in the camera coordinate system with

$$P_{3D}(u,v,d) = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} \frac{(u-c_u)\cdot T}{d} \\ \frac{(v-c_v)\cdot T}{d} \\ \frac{f\cdot T}{d} \end{pmatrix} \tag{8}$$

where u, v and d are the image coordinates and the corresponding disparity value, T and f the baseline and the focal length out of the camera calibration data and X, Y and Z the resulting coordinates of the 3D-Point in the camera coordinate system. In the next step, each 3D point is checked whether it is within a certain three-dimensional clearance profile around the actual course of track and it is deleted if not. After this step, only 3D points within this clearance profile remain in the disparity image.

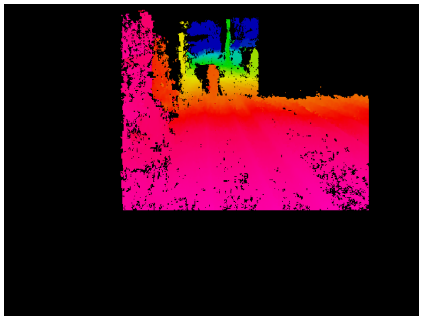
5.2. Labeling

After clearance filtering, the disparity image still contains outliers caused by isolated wrong matches or noise in the input images. The labeling process groups connects adjacent pixels with similar disparity to labels [13] which have to pass certain checks. The first check deletes each label that contains fewer pixels than a certain threshold. The next check calculates the real world size of the label. If the size is smaller than a given minimum size, the label is deleted, too. The remaining labels are declared and reported as obstacles to the train control system. Fig. 14d shows the remaining labels after that process and it can be seen that outliers have been eliminated and pixel that correspond to the obstacle remain in the image.

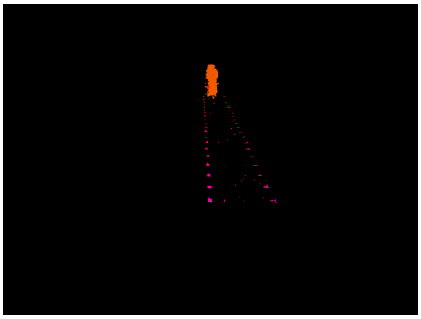
5.3. Dynamic Regions of Interest

The a-priori knowledge of track trajectory allows us to know which parts of the current input images are relevant for obstacle detection. In order to save computation time, only these parts of the images are actually used for depth calculation. Therefore, certain regions of interests (ROI) are independently estimated for each baseline and only for these ROIs, stereo matching is done. In the left rectified image in Fig. 14a two white rectangles indicate the calculated ROIs for BL1 and BL2, respectively.

(a) Left rectified image with ROIs and box around detected obstacle



(b) ROI Depth image



(c) Clearance filtered image

(d) Labeling filtered image

Figure 14: Obstacle detection example

Configuration	Baseline 1			Baseline 2		
	HS	II	OR	HS	II	OR
Reference	1	L-R	1200×660	-	-	-
1	3	L-M	600×330			
	2	L-R	600×330	2	L-M	270×149
	1	L-R	1200×660	1	L-M	540×297
2	3	L-M	600×330			
	2	L-R	600×330	2	L-M	400×220
	1	L-R	1200×660	1	L-M	800×440
3	3	L-M	700×385			
	2	L-R	700×385	2	L-M	500×275
	1	L-R	1400×770	1	L-M	1000×600

Table 2: System configurations, HS = Hierarchical step, II = Input image, OR = Output resolution, L-R = Left and right input image, L-M = Left and middle input image

6. Evaluation Results

In this section our proposed stereo vision based obstacle detection is evaluated on extensive amount of real-world data. We show the detection rate of obstacles on the track including false-positive detections, and we also present performance measurements with various performance enhancements.

To evaluate the obstacle detection abilities of this system, various sequences with different scenarios have been recorded including a fifty minutes sequence with more than 12000 frames of almost the whole 15 km long test track, which starts at WGS84 coordinates (47.999193N;13.920281E) and ends at (47.915449N;13.803991E). A ground truth information of this sequences has been generated manually by categorizing frames whether they contain obstacles on the track or not.

As the parameter space in our proposed system is considerably large (image resolution, number of hierarchical steps, labeling parameters, ...) there exists an almost unlimited number of different configurations that can be applied. According to the needs of the application we chose three configurations alongside the reference configuration of 2.

Table 2 lists the four configurations that are compared, where "Reference" is the reference configuration as it is described in Tab. 1. The three other configurations use the two baseline approach a hierarchical mode setup

Sequence	Frames	Frames with obstacles $\leq 80\,\mathrm{m}$	Description
A	12022	506	Sequence 1 contains almost the whole test track. Obstacles in this sequence are persons on the track and other approaching trains.
B	813	311	Sequence 2 shows a crouching person on the track as the train approaches. Other persons are crossing a crossway.

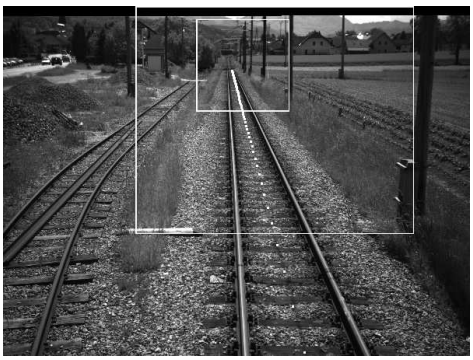
Table 3: Recorded sequences and test cases for system evaluation

according to Fig. 13. The only difference between these configurations is the resolution of the output disparity images.

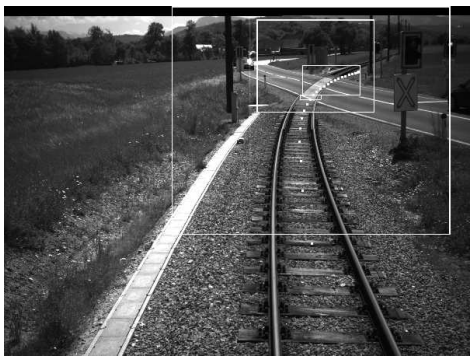
6.1. Results in Obstacle Detection and False/Positive Detection rates

For evaluating the abilities of the system in obstacle detection we have chosen two representative test sequences with various numbers of frames, situations and obstacles. Table 3 gives a short overview of these sequences.

In Table 4 the results of obstacle detection are shown. In sequence A all obstacles at a distance between 10 m and 80 m have been detected by our proposed system with all three configurations. The false positive rates were also low, where the configurations with a higher output image resolution showed the best results. The *filtered* false positive rate indicates frames whose predecessor and/or successor reports a false positive detection, too – i.e, the wrong result has been "confirmed". This rate has more relevance to us, as the test system for autonomous train control is designed to only set actions when an obstacle appeared in at least two subsequent frames. The false positive rates in sequence B are even lower, but on the other hand, there have been a few frames with an obstacle on the track, where nothing has been detected. One possible cause for these errors are inaccuracies in the currently submitted course of track trajectory, because also trains have an elastic suspension that may cause pitch and roll movements of the vehicle. This results in deviations of the trains coordinate system, which are currently not captured properly by the sensor equipment. Fig. 15a shows a track



(a) Accurate track information



(b) Inaccurate track information causing false positive detections

Figure 15: Example of accurate and inaccurate track information

segment, where the provided track information is accurate and matches with the current view (the track information is indicated by a dotted line), whereas Fig. 15b shows a typical case, where the track information deviates from the current view. The virtual track trajectory "dives" into the ground plane and this results in several false positive detections. For the future course of the project we intend to overcome this problem by integrating a track detection in our system, like it is proposed in [14], to compensate these errors.

Another cause for false positives are inaccuracies of the self localization and the recorded track data map. Both are based on GPS measurements and the do have certain inaccuracies and uncertainties (Fig. 16). Again, track detection or even more a SLAM-based approach that uses 3D sensor data could improve self localization and support more reliable obstacle detection.

6.2. Results in Performance

The latency between image acquisition and obstacle report is of high relevance as this time span is a measure for the overall response time of the system. The shorter this time span the faster the train control system can initiate actions to avoid a collision with an obstacle on the track. In this section, analysis results of the computational performance are presented.

Performance analysis has been done for all configurations on the basis of one frame that we consider representative for the vast majority of scenes. The test platform was a standard PC comprising a i7 CPU @ 3.06 GHz and 4×2 (hyperthreading) cores. Beside of the size of the ROIs, almost all parts

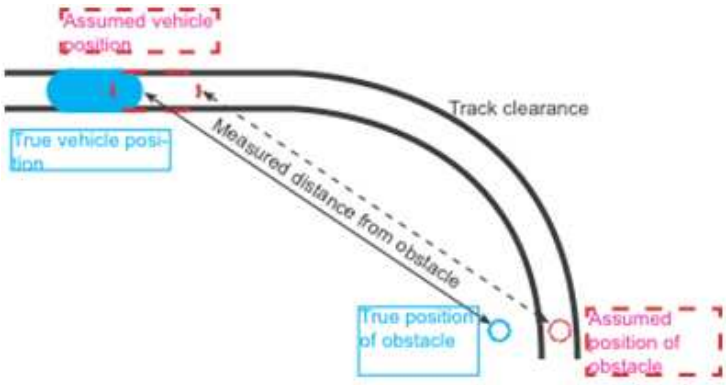


Figure 16: Error in position of obstacles

Sequence	Configuration Number	Obstacle detection rate	False-Positives	Filtered False-Positives
A	Reference	506 (100%)	246 (2.04%)	184 (1.53%)
A	1	506 (100%)	265 (2.20%)	157 (1.31%)
A	2	506 (100%)	167 (1.39%)	94 (0.78%)
A	3	506 (100%)	129 (1.07%)	66 (0.55%)
B	Reference	278 (89.39%)	3 (0.36%)	2 (0.25%)
B	1	306 (98.93%)	28 (3.44%)	6 (0.73%)
B	2	304 (97.75%)	26 (3.20%)	6 (0.73%)
B	3	304 (97.75%)	10 (1.23%)	2 (0.25%)

Table 4: Obstacle detection rate and False/Positive rate

Configuration	Reference	1	2	3
Stereo matching BL1	325.80	48.18	45.54	64.60
Stereo matching BL2	-	24.02	57.35	86.92
Stereo matching total	325.80	72.20	102.89	151.52
Stereo matching FPS	3.07 fps	13.85 fps	9.72 fps	6.60 fps
Combine Disparity Images*	-	18.27	18.72	25.13
Clearance Filtering*	40.90	24.91	23.35	30.06
Labeling*	6.54	6.48	5.73	8.07
Depth image calculation	8.68	6.32	6.50	8.73
Image acquisition	35.00	35.00	35.00	35.00
Total latency time	**416.92**	**163.18**	**192.19**	**258.51**

Table 5: Performance table (all times in ms), *not yet performance optimized

of the software have a constant execution time which is not dependent on the content of the input images. Timing results are listed in Tab. 5, where not only the total calculation time per frame is shown, but also the time consumption of several other stages, and the stereo matching itself is split for each baseline. Image acquisition time has been estimated empirically, it mainly consists of the exposure time and the data transfer time to the host computer over gigabit ethernet. The reference configuration does not use hierarchical stereo matching, ROI mode and slanted correlation masks. It shows the slowest overall performance and especially the slowest performance in stereo matching. As expected, configuration one is the fastest configuration and could deliver stereo matching frame rates of almost 14 fps. Some modules still have potential for performance optimization, e.g., ROIs processing and clearance filtering. With further optimizations we expect even higher frame rates.

7. Discussion and Conclusions

With the current implementation of the stereo vision based obstacle detection system it could be shown that it is possible to fulfill the requirements for this application. The system has successfully detected every obstacle in the recorded test sequences. With the proposed software enhancements a latency lower than the requested 300 ms has been achieved even at an increased resolution of 1400 pixel image width, which could enable a detection

range longer than 80 m.

The results of the system depend strongly on the accuracy of the delivered track information. When the track information is inaccurate, the system may fail in certain situations. For future development it is intended to make the stereo vision based sensor system more independent of the provided track information by, for example, integrating an additional track detection system into the software framework, to enable a correction of the track information or, in the best case, to be independent from any external information.

Another improvement that is intended in the future course of this project is to apply a stereo vision system with thermal infrared cameras, as the obstacle detection system should ideally work reliably in all weather and light conditions. First experimental results are very promising.

[1] Nedevschi, S., Danescu, R., Marita, T., Oniga, F., Pocol, C., Sobol, S., Tomiuc, C., Vancea, C., Meinecke, M., Graf, T., To, T.B., Obojski, M.: A sensor for urban driving assistance systems based on dense stereovision. In: Intelligent Vehicles Symposium, 2007 IEEE. (2007) 276 –283

[2] Labayrade, R., Aubert, D., Tarel, J.P.: Real time obstacle detection in stereovision on non flat road geometry through "v-disparity" representation. In: Intelligent Vehicle Symposium, 2002. IEEE. Volume 2. (2002) 646 – 651 vol.2

[3] Ventroux, N., Schmit, R., Pasquet, F., Viel, P.E., Guyetant, S.: Stereovision-based 3d obstacle detection for automotive safety driving assistance. In: Intelligent Transportation Systems, 2009. ITSC '09. 12th International IEEE Conference on. (2009) 1 –6

[4] Daily, R., Travis, W., Bevly, D.M., Knoedler, K., Behringer, R., Hementsberger, H., Kogler, J., Kubinger, W., Alefs, B.: Sciautonics-auburn engineering's low-cost, high-speed atv for the 2005 darpa grand challenge. Journal of Field Robotics **23** (2006) 579–597

[5] Kadiofsky, T., Weichselbaum, J., Zinner, C.: Off-road terrain mapping based on dense hierarchical real-time stereo vision. ISVC2012 8th International Symposium on Visual Computing (2012)

[6] Humenberger, M., Zinner, C., Weber, M., Kubinger, W., Vincze, M.: A fast stereo matching algorithm suitable for embedded real-time systems. Computer Vision and Image Understanding **114** (2009) 1180–1202

[7] Zinner, C., Humenberger, M., Ambrosch, K., Kubinger, W.: An optimized software-based implementation of a census-based stereo matching algorithm. In: Advances in Visual Computing, Lecture Notes in Computer Science. Volume 5358., Springer (2008) 216–227

[8] Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International Journal of Computer Vision **47** (2002) 7–42

[9] Scharstein, D., Szeliski, R.: (Middlebury Stereo Evaluation Ranking) `vision.middlebury.edu/stereo/`.

[10] Stein, A.N.: Attenuating stereo pixel-locking via affine window adaptation. In: In IEEE International Conference on Robotics and Automation. (2006) 914–921

[11] Bleyer, M., Rhemann, C., Rother, C.: Patchmatch stereo - stereo matching with slanted support windows. In: British Machine Vision Conference 2011. (2011) 1–11 Vortrag: British Machine Vision Conference BMVC 2011, Dundee; 2011-08-29 – 2011-09-02.

[12] Hung, Y.P., Chen, C.S., Hung, K.C., Chen, Y.S., Fuh, C.S.: Multipass hierarchical stereo matching for generation of digital terrain models from aerial images. Machine Vision and Applications **10** (1998) 280–291 10.1007/s001380050079.

[13] Shapiro, L.G., Stockman, G.C.: Computer Vision. 1st edition edn. Prentice Hall (2001)

[14] Gschwandtner, M., Pree, W., Uhl, A.: Track detection for autonomous trains. Advances in Visual Computing: 6th International Symposium, (ISVC 2010) (2010) 19–28