

# Disparity Space Image's Features Analysis for Error Prediction of a Stereo Obstacle Detector for Heavy Duty Vehicles

Alberto Broggi, Stefano Cattani, Elena Cardarelli, Brad Kriel, Michael S. McDaniel, Hong Chang

**Abstract**—Vision-based perception has been explored as low-cost, flexible technology for industrial applications and ADAS. Its inherent flexibility presents a challenge quantifying performance and often even quantifying increases or decreases in system performance as conditions change. Experience enables designers to employ various "rules of thumb" while commercially viable products require quantitative performance.

This paper explores the correlation between features and characteristics of the Disparity Space Image (DSI) and resulting performance for an object detection application. The specific application is an object detection system suitable for highly chaotic environments often found in earthmoving industry. Features and characteristics with strong correlations can be used to improve system design and predict system performances at run-time.

High-quality stereo images are used to characterize baseline system performance. These images are then artificially degraded to simulate fog, darkness, and blurring and subsequent system performance compared to baseline results.

## I. INTRODUCTION

Stereo-vision based perception systems have been developed for industrial and automotive applications and more recently in off-road applications [1]. Vision system maturity in industrial applications is quite advanced where the harsh environments are consistent and controllable. Prototype automotive vision systems have demonstrated remarkable capability but commercial solutions must meet very challenging and uncontrolled environmental variables. The number of parameters available to designers of vision-based systems make these systems adaptable to a wide range of applications. The resulting systems are very complex with direct and indirect feedback paths. Experienced developers can subjectively assess the underlying images and intuitively know how well the system is likely to perform. Translating "rule of thumb" and qualitative assessment into quantifiable parameters will help define the various curves and bounds of the system.

A prototype object detection system based on stereo-vision was jointly developed and tested by VisLab and Caterpillar engineers for off-road environments. The disparity engine generates a 3-D data set, the Disparity Space Image (DSI) [2] or disparity map, based on perspective differences between two images of the same scene. Additional processing of the

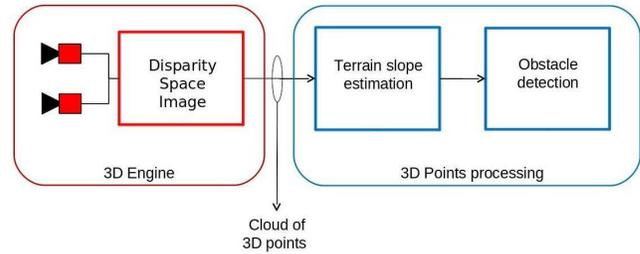


Fig. 1. The block diagram of a stereo-vision based terrain mapping and object detection system.

disparity map yields an estimation of the terrain and non-terrain objects [3]. This is shown in Fig. 1. The prototype system has performed quite well in limited testing. What has been observed in testing is that the performance of the object detection algorithm is strongly correlated with subjective assessment of the quality of the disparity map. Disparity maps with dense, low-noise points produce good object detection results while sparse and/or noisy disparity maps resulted in unacceptable object detection results. Based on these observations, the team looked for quantitative links between detection results and characteristics of the disparity map. As an example, Fig. 2 shows the results of a scene recorded at dramatically different lighting conditions and the resulting disparity maps.

Here it is important to stress an aspect of our approach: in literature already exist some studies about stereoscopic image quality measurement. Image quality metrics based on peak-to-noise-ratio (PNSR) or signal-to-noise-ratio (SNR) [4] or strictly subjective evaluation [5] are not reliable indicators of whether the disparity map quality is sufficient for this specific application. Our approach claims the existence of a link, a function, that puts in a direct relation the DSI and the overall algorithm performance, obtained when feeding the terrain estimation and obstacle detector block with the DSI's 3D points cloud.

This link is formalized as a *linear function* that puts in relation a set of disparity image's *features* and the corresponding results. Once the coefficients of this linear relation are known, the generated metric will also allow us to *predict* the algorithm performances under other conditions.

Studying the correlation between features and errors we

A. Broggi, S. Cattani and E. Cardarelli are with VisLab - Dipartimento di Ingegneria dell'Informazione, Università di Parma, Italy broggi, cattani, cardar}@ce.unipr.com

B. Kriel, M. S. McDaniel and H. Chang are with Caterpillar Inc. {Kriel\_Brad, McDaniel\_Michael\_S, Chang\_Hong}@cat.com

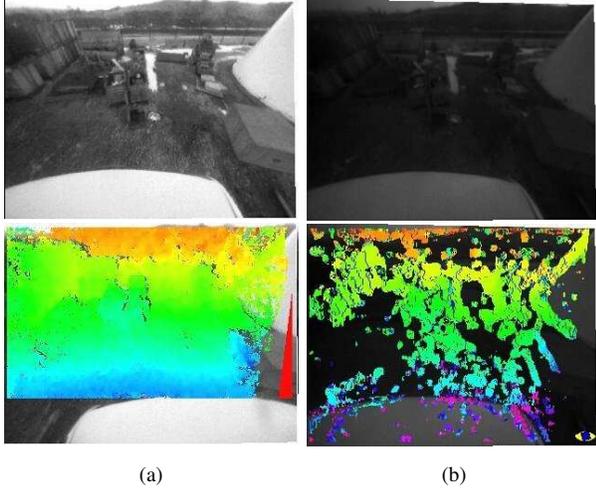


Fig. 2. Here is an example of which are the effects of bad lighting on the Disparity Space Image. On the left column (a) are visible Right images taken under good lighting and the corresponding dense DSI. On the right column (b), the same exact scene as seen earlier in the morning: now the DSI is significantly less dense.

expect also to reduce the feature sets to those most effective in characterizing the disparity quality.

In summary, in this paper defines parameters of interest in the disparity map. Degradation of image quality consistent with darkness, fog, and blur are described. Analysis consists of training a linear predictor using a training set of images and then estimating corresponding performance on a different test set, as the images are degraded. A summary of the correlation between parameter and overall performance is presented along with proposed extensions and research.

The rest of this paper is organized as follows: in Section II a brief introduction to image degradation is provided; in Section III is described which features are used and how they are computed; in Section IV is presented how each feature set can be associated to the corresponding resulting detection error by a linear function and the effect of correlation analysis in reducing the input space; Section V and Section VI present, respectively error prediction results and paper's conclusions.

## II. DEGRADED IMAGES

We used 3 different types of degraded images (Fig. 3(A)(B) and (C)). The degradation was *artificially added*: starting from very good images, taken in good lighting conditions, we added disturbances like noise, atmospheric effect, fog, darkness and sensor noise.

- **Artificial Gaussian blur:** Motion artifacts and poor camera focus result in blurred images. This degradation is approximated using an artificial gaussian blur where the original imaged is convolved with this function:

$$G(u, v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}} \quad (1)$$

values of  $\sigma$  goes from 1 to 50, increasing of 1 steps. The convolution matrix size goes from 1 to 40.

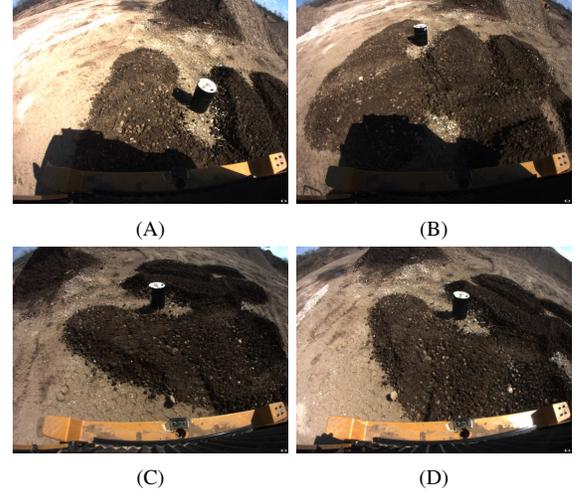


Fig. 3. Image object of degradations: (A), (B), and (C) are part of the training set; (D) is the test set.

- **Artificial Fog:** Atmospheric fog reduces contrast in a scene proportional to the distance between the observer and the point in the scene. Artificial fog is introduced to the original images using the following equations (OpenGL model [6]):

$$fogged\_pixel(u, v) = pixel(u, v) \cdot f + fogcolor \cdot (1 - f) \quad (2)$$

where  $f$  is the blending factor, computed using an exponential function:

$$f = e^{-d \cdot z} \quad (3)$$

where  $d$  is the *fog density* and  $z$  is the eye-coordinate distance between the viewpoint and the fragment center. Fog density goes from 0 to 1, fog color is (127, 127, 127) *RGB*.

- **Artificial darkness and noise:** Low ambient light results in two phenomenon, reduced image brightness and increased sensor noise. Darkness is simulated using an attenuation factor coupled with an independent, additive gaussian noise.

$$att\_pixel(u, v) = (pixel(u, v) \cdot a + noise(u, v)) \cdot gain \quad (4)$$

where  $a$  is an attenuation factor,  $noise(u, v)$  is the Gaussian white noise added to the signal by the sensor and  $gain$  represents the overall signal amplification applied by the camera. The system automatically adjusts shutter speed and gain to keep the average image brightness within an acceptable range. Consequently, as the attenuation increases, gain increases, amplifying the noise over the signal value. Attenuation range for  $a$  if from 1 to 0; range for  $gain$  is from 0 to 9.32, while noise is Gaussian with  $\sigma$  from 1 to 5. These ranges were computed according with the camera specifications [7].

Each image was taken in good lighting conditions, at known Shutter and Gain values, and then we iteratively

applied different levels of degradations, from the lightest to the strongest. In Fig. 4 some examples of image degradations, and corresponding DSI, are shown.

### III. DSI FEATURES

First of all, this paper does not want to present a comprehensive and general set of DSI features, to be applied in any situation. The paper focus is on the framework: how to link a set of DSI features (however defined) to the algorithm performances. Any custom set of DSI features could be applied to the method. Having said this, lets see which features we used to test our methodology.

As noted in the introduction, some existing methods focus on global measurements. A challenge with global metrics is that performance in some region may differ substantially from the performance suggested by the global metric. While it may be possible to assess the global quality of the image and DSI, subdividing the images into smaller regions and assessing local quality might be more appropriate. Local quality metrics might identify regions where performance is not acceptable that would otherwise be masked when evaluated using global metrics. Equally challenging is the appropriate subdivision process and the choices of size, geometry, and overlap. Even with a relatively small number of parameters and subdividing regions, the population of potential DSI features quickly grew to several hundred members.

Subdividing the DSI into suitable regions is complicated by constantly changing scene characteristics and the non-uniformity that degradations affect various regions within a scene. A natural subdivision might be to segment the DSI into stripes to infer range-based performance. The stripes can be similarly subdivided to produce non-overlapping squares. This approach results in non-overlapping squares of uniform physical dimension. A second natural subdivision might be to segment the pixel space into non-overlapping squares to keep metrics such as number of disparities per region uniform while allowing the range to vary with each region. Because specific scene characteristics such as shadows are not known a-priori, non-overlapping segments might mask the earliest indications of reduced performance. As such another natural subdivision is to convolve a focus region (in this case, a square) with the original DSI to produce a series of overlapping regions.

For this analysis, using a 320x240 disparity space image, we focus on the following subdivisions:

- Stripe (*DSIRange*): 18 non-overlapping 0.5m lateral stripes consistent with the algorithm resolution used to segment the DSI into terrain and objects.
- Square (*DSIFixed*): 103 non-overlapping 20 pixel by 20 pixel squares.
- Window (*DSIMobile*): 186 overlapping squares formed by a 150x150 pixel window shifted 9 pixels at a time.
- Image (*DSIGlobal*): here the whole actual DSI computation area is analyzed; this area might be smaller then the entire image.

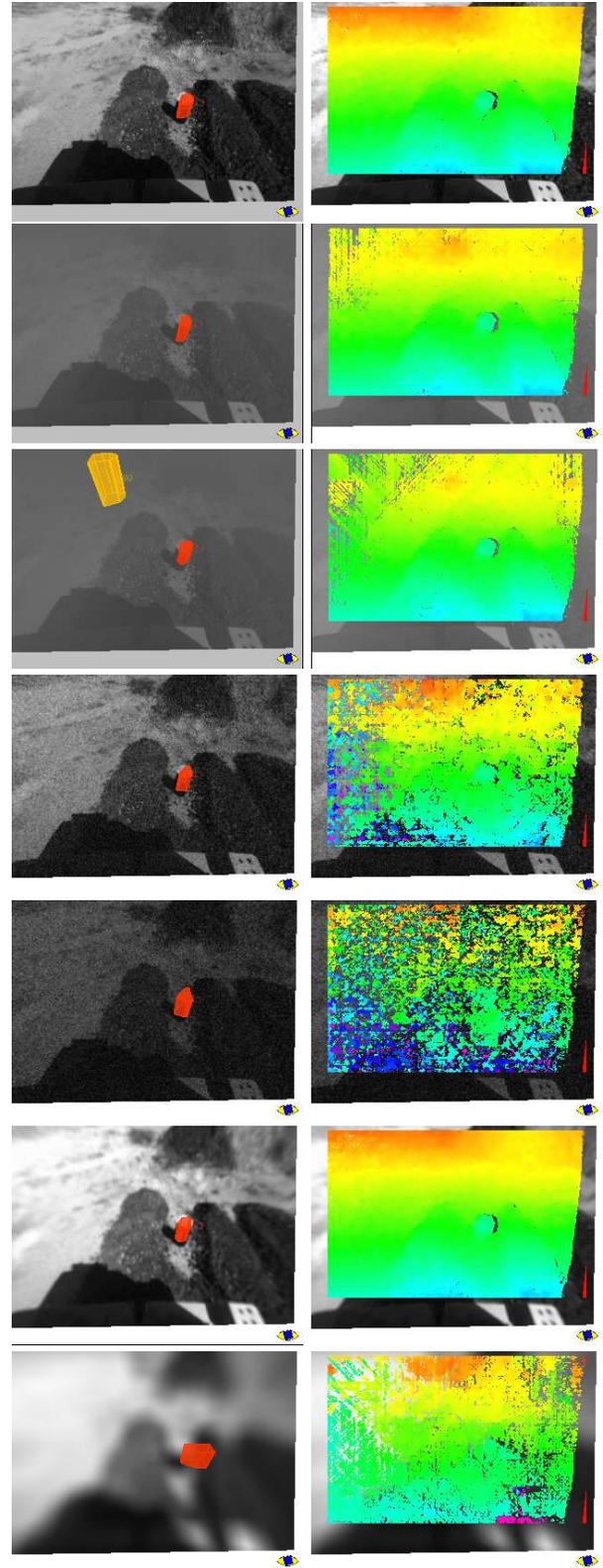


Fig. 4. Some example of image degradations and corresponding DSI. From top: image in perfect lighting conditions; Fog, with density  $d = 0.30$ ; Fog, with density  $d = 0.37$ , first false positive; Artificial Darkness and Noise, with  $attenuation = 93.5\%$  ( $a = 0.065$ ),  $gain = 18.062dB$ ;  $attenuation = 96.5\%$  ( $a = 0.035$ ),  $gain = 22.499dB$ ; Artificial Blur, with  $\sigma = 6$ , matrix size 6;  $\sigma = 50$ , matrix size 50;

Let's now introduce some symbols related to feature calculation:

- $N$  is the number of right image pixels the disparity engine attempts to match with left image pixels. A DSI with 100% density contains  $N$  points. In practice, this is lower than the  $width \cdot height$  of the image.
- $N_{stripe-i}$  is the theoretical number of disparities on the  $i$ -th image stripe, corresponding to a  $0.5m$  wide stripe on the terrain, in case of fully dense DSI.
- $N_{fixed-i}$  is the theoretical number of disparities expected into each the one of the  $20 \times 20$  pixels non-overlapping areas in which the image is divided; considering only the region of interest for obstacle detection, we have 103 fixed areas;
- $N_{mobile-i}$  is the theoretical number of disparities expected into each one of the  $150 \times 150$  pixels overlapping areas in which the image is divided; each area overlaps with its vertical and horizontal neighbor by 141 pixels; we have 186 mobile windows;

To compute the DSI features set within a region, the following characteristics are evaluated:

- *Filling*: density of the disparity image  $\frac{n}{N_e}$ , where  $n$  is the actual number of disparities obtained on the area and  $N_e$  is the corresponding theoretical number of disparities expected;
- *Mean*: average of the obtained disparity on the area;
- *Variance*: disparity variance among the area;
- *Diff*: average of the differences between disparity values on the area and the expected values in case of flat terrain;
- *GapsNumber*: number of disparity gaps (holes), considering only gaps with an area greater than 15;
- *GapsMeanArea*: average gaps area  $\frac{\sum a_{gaps}}{N_{gaps}}$ ;
- *GapsStandardDeviation*: gaps; area standard deviation;
- *GapsFilling*: this feature represents the ratio between gaps' area and the overall image's number of missing disparity:  $\frac{\sum a_{gaps}}{N-n}$ ; it is a measure of the importance of gaps in the overall disparity density;
- *GapsSumDistances*: this feature measure the average distance between each gap and its closest neighbors, normalized with respect of the gap radius; it measure the average gaps' spatial concentration.

The overall number of features is about 840. Since feature values belong to different ranges, data is normalized according to their standard deviation: for each feature type, its standard deviation among different frames is calculated and used as factor for the normalization. This normalization is not performed runtime, but it is necessary only during the training phase.

#### IV. FEATURES ANALYSIS

To remove redundancy and weak informative data in the feature set different methods for features selection [8] [9] has been presented in literature and some comparative studies [10] [11] [12] has been conducted in order to enhance their capabilities.

Filter ([13]) and wrapper ([14]) models are the two principal classes of methods presented: despite accuracy lost, the second approach is more reliable than the first one.

Our analysis tries to find a *linear* relation between the features values and the corresponding detection error in output. This analysis was made on a set of images containing an obstacle detected by the algorithm where the disparity appeared to be very good. Then, for each of these images, the following steps are performed:

- 1) the obstacle is detected and its position  $(x_r, y_r, z_r)$  stored;
- 2) a degradation is applied to the image;
- 3) the obstacle is again detected on the  $i$ -th degraded image, on position  $(x_i, y_i, z_i)$ ;
- 4) the detection error is computed as the distance between baricenters, normalized with the obstacle distance:

$$e_i = \frac{\sqrt{(x_i - x_r)^2 + (y_i - y_r)^2 + (z_i - z_r)^2}}{\sqrt{x_r^2 + y_r^2 + z_r^2}} \quad (5)$$

- 5) return to step 2, increasing the degradation level.

At the end of this process, we will have a vector of errors  $E$  and a matrix of features values  $A$ . The  $A$  matrix has a number of columns equal to the number of features, and a number of rows equal to the number of analyzed images. By the analysis of a sufficiently large set of degraded images, we can solve this problem:

$$Aw = E \quad (6)$$

where  $w$  is the *features' weight vector*. This vector contains 840 weights, one for each feature.

Since the linear system is overdetermined we use the least square fitting to determine the solution corresponding to the best fit of the input data.

Note how we do not record any values in case of absent detection. If the algorithm is not able to detect any obstacle, we do not introduce "default" errors values, in order to avoid the presence of outliers that could affect the linear transformation.

##### A. First features analysis and reduction

In the first phase we computed the whole feature set  $F$  described above for each DSI. After a first analysis of the results, we noticed that:

- the *Filling* features are the predominant;
- the most important features are located around the obstacle or in some *image dependent* regions where the DSI degrades more significantly than the rest of the image (e.g. strong shadows).
- the *Gaps*, *Variance* and *Diff* features are presented with very small weight, or completely absent.

As a consequence, we can conclude that:

- local features are preferable to global ones, as already claimed in the Section III introduction: there are regions of the image that are more sensitive to degradations, their position is unknown and their distinctive characteristics are, virtually, endless and unpredictable;

- *Gaps* features seems to be not significant. Big disparity gaps are for sure a cause of detection errors but, evidently, their number and size are not in linear relation with these errors.
- on the contrary, detection errors seem to be driven by *Filling* features values. Three kinds of Filling feature exist: fixed, mobile and ranges. Mobile are actually too many and too demanding for a real time processing; ranges might not be able to detect vertical degradations; fixed windows appear to be the best trade-off between image coverage and reduction of used features.

In conclusion, the reduced features set  $F_{red}$  is made only of  $DSIFixedFillig$ .

### B. Generalization

As mentioned in the previous Subsection, local features are used to estimate detection errors. However, if a given image is characterized by weak disparity in a specific area, the corresponding local filling feature will have a high weight; but, on a different image, the same region might be not important, as shown in Fig 5.

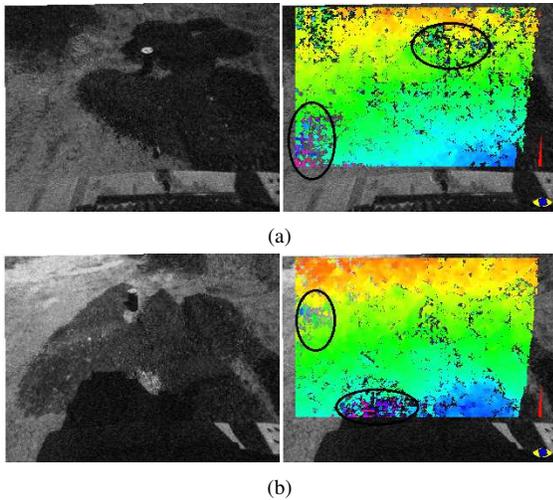
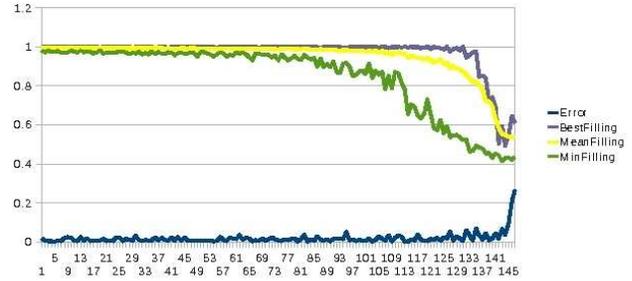


Fig. 5. Example of how the most degraded image regions can change from image to image: (a) test case A; (b) test case C. See also Fig. 3.

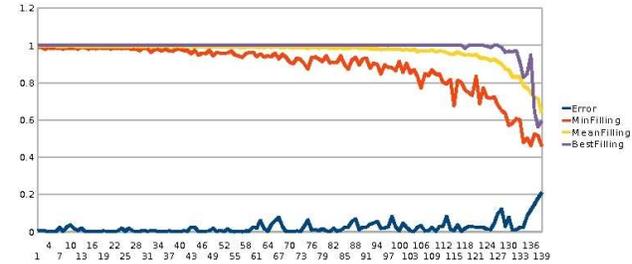
Our idea to overcome this problem has been to use a pair of *functions* of local features:

- *DSI Filling Mean*: this is the average filling of the local 20x20 cells analyzed:  $\frac{\sum DSIFixedFillig_i}{N}$ ;
- *DSI Filling Min*: after having ordered the  $DSIFixedFillig$  features, we take the average value of the *smallest* 3. This value represents a kind of *worst case scenario*: a filling value related with the less filled region of the image.

Testing these new features on our training set, we discovered that their weights  $w$  are almost constant. It seems that the detection error is linked to these two values in a way that do not change from image to image.



(a)



(b)

Fig. 6. Error trend compared to filling average, best filling feature (for the given image) and min filled feature, using two different images: (a) test case A; (b) test case C (see also Fig. 3). All values are percentage: baricentrum error with respect of the obstacle distance; filling percentage. All values are plotted against 150 image degradation levels: Artificial darkness and noise, with Gaussian noise of 5 sigma. The applied brightness attenuation is from 0% to 100%, with steps of 1 until 50%, and with steps of 0.5 from 50% to 100%.

Let's take a look at chart in Fig. 6. Here we can see the error trend compared to *DSI Filling Mean*, best filling feature and *DSI Filling Min* features. In both test case A (a) and B (b) we can notice how:

- 1) there is a feature that follows the error trend better than the others. This feature has been selected and printed on the chart. Unfortunately, this feature is not the same on the two images, but it represents a *DSI* degradation localized in a specific region on each image;
- 2) *DSI Filling Mean* and *DSI Filling Min* are always present, and they follow the error trend in a similar manner on both images;
- 3) the way *DSI Filling Mean* and *DSI Filling Min* follow the error trend is close enough to let us find a mathematical relation between error and features values;
- 4) *DSI Filling Mean* and *DSI Filling Min* anticipate a little the error trend: they start to rise a little in advance, in particular the min filling. This is reasonable, since it represents a worst case scenario.

### C. Final Features weights

At the end of this analysis, on the basis of the considerations 2) 3) and 4) of the previous Section, we concluded that *DSI Filling Mean* and *DSI Filling Min* are the features we are looking for.

Running the algorithm on the whole training set we found the following *weight* values:

Feature	Weight
DSI Filling Mean	-0.307369
DSI Filling Min	-0.174026
(Const)	0.481814

where *Const* is a fictitious feature used to make the linear system in Equation 6 solvable.

## V. RESULTS

Now it is time to verify how the two features selected, and their corresponding weights, are able to predict detection errors. We first trained our system on test cases A B and C, then we tested the error prediction capabilities also on test case D (see Fig. 3).

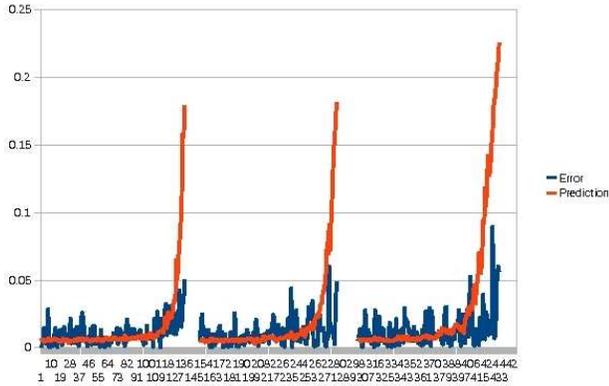


Fig. 7. Error prediction: here we can see the detection errors measured on test case D, compared with the error prediction based on DSI features, both plotted against 450 Artificial Darkness and Noise degradation levels (1-150  $\sigma = 3$ , 151-300  $\sigma = 4$  and 301 to 450  $\sigma = 5$ ).

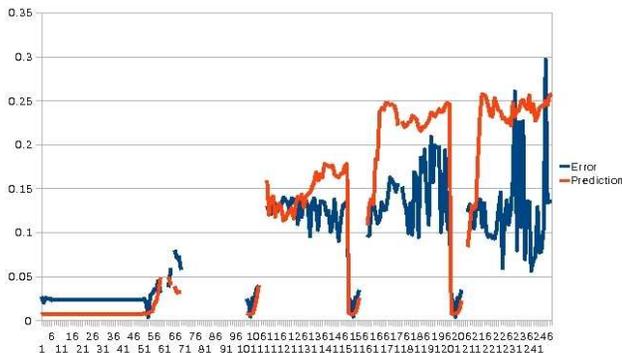
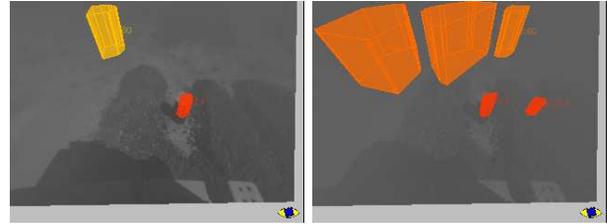


Fig. 8. Error prediction: here we can see the detection errors measured on test case D, compared with the error prediction based on DSI features, both plotted against 250 Artificial Gaussian Blur degradation levels (1-50  $\sigma = 1 - 50$ , radius=1; 51-100  $\sigma = 1 - 50$ , radius=10; 101-150  $\sigma = 1 - 50$ , radius=20; 151-200  $\sigma = 1 - 50$ , radius=30; 201-250  $\sigma = 1 - 50$ , radius=40).



(a) (b)

Fig. 9. Wrong detections in case of fog: note how the false detections start to appear first in the furthest area, where the fog is more dense.

In Fig. 7 we can see the detection errors obtained on test case D while applying a Artificial Darkness and Noise degradation (3,4 and 5  $\sigma$ ) compared with the error prediction based on the *DSI Filling Mean* and *DSI Filling Min* features, using weights we found in Table IV-C. The predicted error trend follows the actual error trend, while the values tend to diverge in correspondence of the end of each sequence. This should not be surprising. Consider what happens around those parts of the chart where no errors values were recorded, that correspond to the first appearance of false negatives: here the algorithm was not able to detect the obstacle at all. Hence, no error values are reported, since it is not measurable. Filling, on the contrary, is still measurable, and reaches very low values, leading to high values of predicted errors. In other words, the malfunctioning here does not consist in errors while estimating the obstacle position, but in missed detections. Hence, in general, we can *interpret* the overestimation as sign of some malfunction, including, high errors or missed detections.

In Fig. 8 we can see the error prediction based on the same features, compared with detection errors obtained when applying Artificial Gaussian Blur degradation on test case D. Again, the general error trend is followed, but we can observe a certain overestimation after frame 120. After this point the estimated error tends to rise quickly as soon as the blurring  $\alpha$  become significant. This can be explained looking what happens in the last example of Fig. 4: here the blurring causes low levels of filling especially in the upper left corner. This is case of *localized* degradation. Consequently, since the error predictor does not know where are the obstacles (if any), it provides an error estimation for the *worst case scenario*, as well explained in Sec IV-B when talking about *DSI Filling Min* feature.

### A. The Fog case

Fog represents a special case: it does not reduce significantly the disparity density. The main effect of fog is to *change* the disparity values: fog tends to *homogenize* the image, canceling edges and reducing sharpness; consequently the values into the DSI have nothing in common with the actual objects and 3D terrain characteristics. The result is the detection of false obstacles (see Fig. 9).

In Fig. 10 we can see the actual detection errors (Error), in case of fog, compared with error prediction based on DSI

features (Prediction N). Here the relation is less evident and strong. It seems that, around frame 30, the disparity density slightly decreases and it remains constant until the end of the simulation.

In Fig. 10 we can see the error prediction (Prediction F) obtained using, as a distinctive feature, *DSIRangeDiff* and *DSIRangeMean* of the last stripe (see Section III). We can notice how the prediction works much better than before. This confirms the previous analysis: progressively adding fog does not cause strong disparity density changes, but it just modifies the disparity mean value. Unfortunately, the mathematical linear relation between *absolute* disparity values trend and the detection error trend are strongly image dependent: disparity ranges depend on 1) calibration parameters 2) the specific scene, made of terrain slope and obstacles.

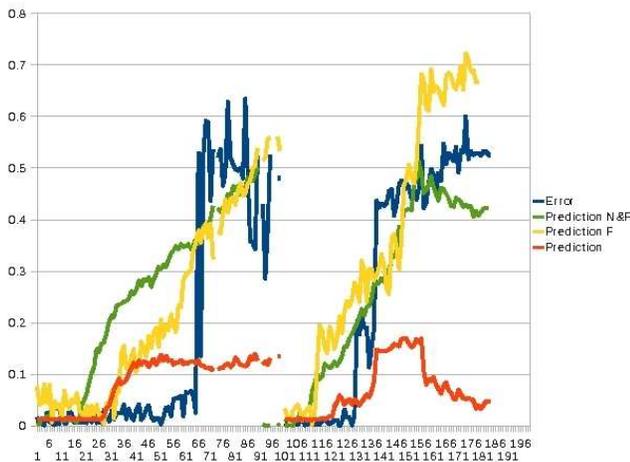


Fig. 10. Error predictions with fog images: here we can see the detection errors measured on test case A (0-100) and B (101-200), applying fog degradation, compared with the error prediction based on DSI features. Prediction N is based on the weight found in Sec. IV-C; Prediction F is based on fog features; Prediction N&F is based on mix filling features and fog features.

In Fig. 10 we can see the error prediction (Prediction N&F) obtained on the fog images using a mix filling features and fog features. The results are still good, since, on these images, the value of filling features is very low, so the prediction is dominated by the fog features' values. Similar result would be obtained the other way around, trying to predict error using both types of features on the images we used to train the filling features: here the dominating values would be the filling ones, leading again to a good prediction performance.

## VI. CONCLUSIONS

This document is aimed at discovering relations between some of the DSI properties and obstacle detection errors obtained when the DSI is used in the perception system developed by VisLab and Caterpillar and introduced in Section I. At the end of this analysis we can conclude:

- there exists a *linear* relation between *global disparity density*, *minimum local disparity density* and detection error, with coefficients shown in Table IV-C;

- since we used a mix of local and global features, the error estimation represents a *worst case scenario* (i.e. obstacle lays exactly in that degraded part of the image);
- there are some types of degradations that do not significantly affect the disparity density, like fog. They just change the values of the disparities, hence it is very difficult to detect them without posing any assumptions on terrain slope and obstacles size and position.
- in fog images there exists a relation between detection errors and *disparity values*; however this relation is very dependent on the specific images and/or the stereo system used.

The prediction error computation time is about 670  $\mu$ s on a 1.86GHz Intel Core 2 with 1.96 of RAM.

## ACKNOWLEDGMENT

The authors gratefully acknowledge Caterpillar Inc. for the support to this research.

## REFERENCES

- [1] Defence Advanced Research Projects Agency (DARPA), "Urban Challenge 2007," Available <http://www.darpa.mil/grandchallenge>.
- [2] Mirko Felisa and Paolo Zani, "Incremental Disparity Space Image computation for automotive applications," in *Procs. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, St.Louis, Missouri, USA, Oct. 2009.
- [3] Andrea Cappalunga, Stefano Cattani, Alberto Broggi, Michael S. McDaniel, and Susmita Dutta, "Real Time 3D Terrain Elevation Mapping Using Ants Optimization Algorithm and Stereo Vision," in *Procs. IEEE Intelligent Vehicles Symposium 2010*, San Diego (CA), USA, June 2010, pp. 902-909.
- [4] Jiachen Yang, Chunping Hou, Yuan Zhou, Zhuoyun Zhang, and Jichang Guo, "Objective quality assessment method of stereo images," in *Proc. 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video*, May 2009.
- [5] Xu Wang, Mei Yu, You Yang, and Gangyi Jiang, "Research on subjective stereoscopic image quality assessment," in *Proc. SPIE 7255, 725509 (2009)*, 2009.
- [6] Bo Sun, Ravi Ramamoorthi, Srinivasa G. Narasimhan, and Shree K. Nayar, "A practical analytic single scattering model for real time rendering," in *ACM SIGGRAPH 2005 Sketches*, New York, NY, USA, 2005, SIGGRAPH '05, ACM.
- [7] Allied Vision Technologies, "Stringray Technical Manual," Available <http://www.alliedvisiontec.com/>.
- [8] Jihong Liu and Guoxiong Wang, "A hybrid feature selection method for data sets of thousands of variables," in *Advanced Computer Control (ICACC), 2010 2nd International Conference on*, 2010, vol. 2, pp. 288-291.
- [9] Chris Ding and Hanchuan Peng, "Minimum redundancy feature selection from microarray gene expression data," *Computational Systems Bioinformatics Conference, International IEEE Computer Society*, vol. 0, pp. 523, 2003.
- [10] Yiming Yang and Jan O. Pedersen, "A comparative study on feature selection in text categorization," 1997, pp. 412-420, Morgan Kaufmann Publishers.
- [11] Monica Rogati and Yiming Yang, "High-performing feature selection for text classification," 2002, pp. 659-661.
- [12] Mehdi Pirooznia, Jack Yang, Mary Q. Yang, and Youping Deng, "A comparative study of different machine learning methods on microarray gene expression data," *BMC Genomics*, vol. 9, no. Suppl 1, 2008.
- [13] George H. John, Ron Kohavi, and Karl Pfleger, "Irrelevant features and the subset selection problem," in *MACHINE LEARNING: PROCEEDINGS OF THE ELEVENTH INTERNATIONAL*. 1994, pp. 121-129, Morgan Kaufmann.
- [14] Ron Kohavi and George H. John, "Wrappers for Feature Subset Selection," *Artificial Intelligence*, vol. 97, no. 1-2, pp. 273-324, 1997.