2018 21st International Conference on Intelligent Transportation Systems (ITSC)
Maui, Hawaii, USA, November 4-7, 2018

Real-time Stereo Reconstruction Failure Detection and Correction using Deep Learning

Vlad-Cristian Miclea, Liviu Miclea and Sergiu Nedevschi

Abstract— **This paper introduces a stereo reconstruction method that besides producing accurate results in real-time, is capable to detect and conceal possible failures caused by one of the cameras. A classification of stereo camera sensor faults is initially introduced, the most common types of defects being highlighted. We next present a stereo camera failure detection method in which various additional checks are being introduced, with respect to the aforementioned error classification. Furthermore, we propose a novel error correction method based on CNNs (convolutional neural networks) that is capable of generating reliable disparity maps by using prior information provided by semantic segmentation in conjunction with the last available disparity. We highlight the efficiency of our approach by evaluating its performance in various driving scenarios and show that it produces accurate disparities on images from Kitti stereo and raw datasets while running in real-time on a regular GPU.**

I. INTRODUCTION

Depth perception is a very important problem in autonomous driving. Stereo reconstruction is the traditional method for depth measurement providing very accurate solutions at relatively low cost.

Stereo reconstruction algorithms have been traditionally classified as being either local or global. Local methods rely on a small support window over which a similarity criterion is applied. On the other hand global methods compute the disparity of all pixels in the image by optimizing a global energy function. One of the most reliable stereo methods is the Semi-Global Matching (SGM) [1]. SGM falls in between local and global categories, ensuring close-to global consistency while consuming a reasonable amount of resources. The method approximates a 2D energy minimization by several 1D optimizations. State of the art provides various SGM implementations, on different platforms CPU [2], GPU [3] or FPGA [4], most of them obtaining real-time performances.

With the increased prosperity of deep learning, convolutional neural networks have been lately used for computer vision tasks such as depth computation. CNNs enable methods dealing with stereo cost computation [5], optimization [6], post-processing [7], end-to-end stereo [8] [9], depth upsampling [10] [11] or depth estimation from single image [12].

Although stereo reconstruction is a well-studied problem, most of the approaches in literature only concern with producing higher accuracy percentages on disparity datasets such as Kitti [13] or Middlebury [14], few approaches concerning about the robustness of the solution. Since sensors in cameras are error-prone, an accurate solution that fails in critical situations might be extremely expensive, especially in the context of autonomous driving.

The inherent failures that have been met in practical usage of stereo, in conjunction to the apparition of LiDAR (Light Detection and Ranging), a more robust and trustworthy sensor, lead to a decrease in usage of stereo. Although extremely accurate for depth measurements, LiDAR has the disadvantage of being more expensive and of giving sparse results.

In order to overcome a part of the stereo shortcomings we tackle here the problem of stereo reconstruction failure detection and correction. We initially present a method that detects the most common types of short-term failures in stereo. Consequently, we modify the stereo pipeline such that it also detects faults by adding several testing points. Furthermore, we introduce a novel stereo camera error concealment method that mitigates the error previously found, producing an accurate depth map in real-time. Our method uses deep learning techniques to compute a novel disparity map by using depth information from previous frame as well as semantic and RGB information from current frame.

The paper starts with presenting the state of the art in stereo reconstruction and possible solutions to overcome errors caused by camera sensors in the stereo context. It continues with presenting the most relevant stereo failures, several possible ways of detection and a modified traditional stereo pipeline that incorporates these detection methods. Section 4 introduces a novel disparity prediction method based on convolutional neural networks (CNNs). In section 5 we present a thorough evaluation of our method and we discuss the improvements given by our method in various driving scenarios. Finally, we conclude the paper in section 6.

II. RELATED WORK

A. Classic Taxonomy for Stereo Reconstruction

Although lately state of the art benchmarks show that end-to-end deep learning-based stereo methods such as [8] or [9] produce the most accurate results, they need a large amount of data for training and most of them can not run in real time. Moreover, we are not 100% sure how such methods behave when dealing with an unrecognizable situation, that has not previously met in training. Therefore, we must rely on traditional stereo taxonomy proposed by D. Scharstein and R. Szeliski [14] that divides the stereo problem into four

The authors are with the Faculty of Automation and Computer Science, Technical University of Cluj-Napoca, Cluj-Napoca, RomaniaE-mails: Vlad.Miclea@cs.utcluj.ro, Liviu.Miclea@aut.utcluj.ro, Sergiu.Nedevschi@cs.utcluj.ro

978-1-7281-0323-5/18/$31.00 ©2018 IEEE

main phases – cost computation, aggregation, optimization, refinement, each phase being responsible for solving a particular sub-problem.

Cost computation is generally computed by using a specific metric to find similarities between patches from left and right images. Traditional metrics are either intensity-based – SAD, SSD, NCC [15], non-parametric – Rank Transform, Census Transform [16] or based on extracted features. These can be either hand-crafted [17] [18] or learnable [5] [19] [20].

Aggregation generally is done by adding additional pixels from a support window. Best solutions for this step usually rely on accurate edge detection, object edges being used for setting the aggregation window boundaries [21]. Aggregation is followed by the optimization step, in which global consistency of the disparity is assured. Methods such as SGM [1], Graph Cuts [22] or Belief propagation [23] are used for this step. Finally, a refinement of the disparity is performed, generally by filtering with median [24], bilateral [25], guided [26] or even learnable filters [7].

B. Missing frame recovery for stereo

One of the top methods dealing with error concealment for frame misses in the context of stereo has been developed by Chen et al. [27]. In this work the authors propose to infer the missing right frame by considering the temporal change detection (from two successive left images) and an estimation of the disparity.

A different method that also recovers right image is proposed by Chung et al. [28]. This method exploits motion vectors of each available frame thus being able to conceal each type of image error. Several other similar approaches have been proposed in literature [29], [30], [31].

Instead of using methods that try to recover the missing frames by classic concealment methods, we apply here a different approach, in which we employ learning mechanisms to generate the disparity map by using convolutional neural networks.

III. Stereo Camera Fault Detection

A. Problem formulation

Let $I_L(t)$ and $I_R(t)$ be two simultaneously captured images from a calibrated stereo system at time t. A traditional stereo method can generate a disparity map $I_D(t)$, from which a depth map can be easily computed. For the next frame in temporal succession – at time $t+1$ – we assume that image I_R becomes either unavailable or incorrect. This could be caused by various specific camera errors or by camera decalibration. The main goal for this work is to overcome the lack of information from second camera in order to generate the disparity $I_D(t+1)$ by using all available information. This process is highlighted in Figure 1.

B. Stereo camera defects

The defects in digital imaging CMOS sensors depend on many causes, such as the physical dimensions of the pixel photosensitive area or the homogeneity of silicon. The variations in the manufacturing process produces pixels with different physical dimensions. Also, the efficiency to convert photons to electrons is given by the inhomogeneity naturally present in silicon [32].

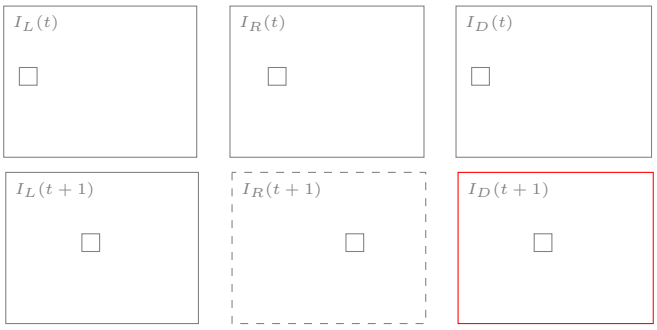


Fig. 1: Problem formulation for stereo error concealment; $I_D(t+1)$ is computed by using relevant info from temporal change in I_L together with info from previous disparity I_D

Because physical defects are obviously not numerable in terms of type, location and time of occurrence, the testing field deals with faults, which model the physical defects to make the numerable, and thus tractable.

Techniques like FMEA (Failure Mode and Effects Analysis), RCA (Root Cause Analysis), and FTA (Fault Tree Analysis) can be used to diagnose a system and to avoid failures [33]. To be able to define the failure modes of a stereo camera (SC), based on its proper operation, we used a FMEA analysis. The failure modes lead to several categories of failures effects. These will be discussed in the following section, in conjunction with the pipeline shown in Figure 2.

FMEA method associates to each failure mode a risk level (RPN Risk Priority Number), and decide the corrective actions for the most severe issues. The application of FMEA method for a SC request the following information: SC component, function, failures, modes of failures, effects of failures, causes of failures, occurrence of failures (OCC), severity of failures (SEV), current control laws, detection of failures (DET), and recommended actions [34]. We used a 10-point ratings of SEV, OCC and DET. A case study for the FMEA analysis is presented in Table I, where three failure types are considered.

C. Detection inserted in stereo pipeline

The main reason for choosing a stereo method that follows the traditional stereo taxonomy is that intermediate results are quantifiable and thus can be evaluated. This introduces the possibility of detecting the faulty results after each step. The workflow of our proposed method can be seen in Figure 2. Although we present here only the case in which the right camera suffers a defect, the detection and concealment methods are viable for the mirror case as well.

The main branch of the workflow uses the most robust methods in each category, such that intermediate results are accurate. A center-symmetric Census metric [16] is preferred for cost computation, followed by cross-based cost aggregation [35], SGM optimization [1] and disparity refinement using a median filter [24].

TABLE I: Failure types for stereo camera – case study

	Failures		
FMEA information	**Fail1**	**Fail2**	**Fail3**
Failure mode (FM)	Right image missing	Image not legible	Image parts not legible
Failure effect (FE)	All pixels are black	Most of the pixels corrupted	Image parts have been corrupted
SEV	7	5	3
Possible cause (PC)	Hardware malfunction (thermal effect)	Large decalibration	Some pixels are faulty
OCC	3	5	7
Current control	Detection using a mask	Lens distortion correction	Integrity verification (forgery det.)
DET	3	2	1
RPN	63	50	21
Det Solution	Check using a mask	Left-right consistency check	Left-right consistency check

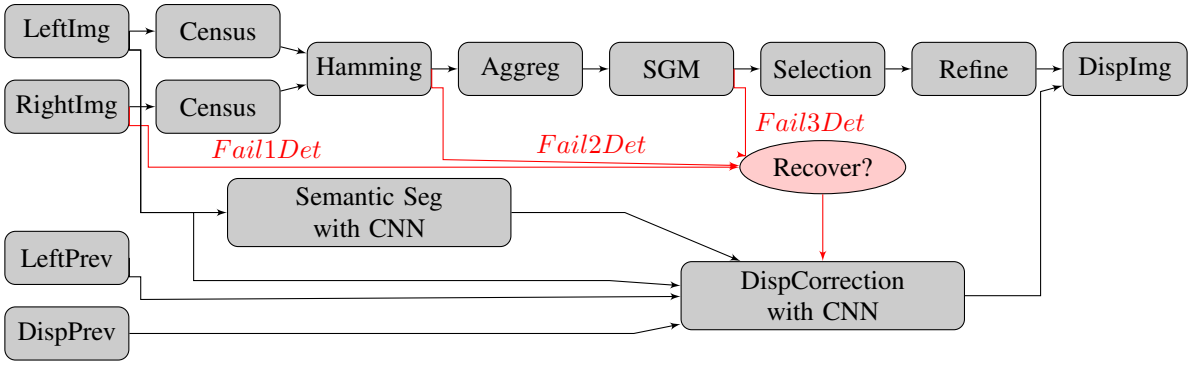


Fig. 2: Workflow of the Proposed Method

Three failing test points are added inside the main pipeline:

1) After image capturing – check if the image has been acquired
2) After cost computation – check if most of the image pixels have been corrupted
3) After optimization – check if several pixels have been corrupted

The initial detection ($Fail1Det$) captures if the right image is missing. This can be easily done by simply verifying if all pixels in the image are black (eg by using a mask).

Following this initial test, cost computation step is performed by using a non-parametric metric. Patches from left and right images are transformed according to the Census transform. Then, a Hamming distance is computed between the two census-transformed patches:

$$C_{comp}(p,d) = Ham(T_l(p, N(p)), T_r(p-d, N(p-d))) \quad (1)$$

where Ham is the Hamming distance, T is the Census Transform of the image patch centered in p and $N(p)$ is the chosen neighborhood.

Once the cost computation has been performed, the second type of faults can be detected ($Fail2Det$). A testing point is inserted to find if most of the pixels in the right image have been corrupted. This can happen due to possible camera decalibrations caused by vibrations. A left-right consistency check is applied over the cost volume. Therefore, we check the left-view disparity map to be equal to the projected right-view disparity map:

$$d_{LR}(x,y) = d_{RL}(x + d_{LR}(x,y), y) \quad (2)$$

where d_{LR} and d_{RL} are two intermediate disparity maps, generated from the initial cost volume (after cost computation). If the number of unreliable points inside the disparity generated after the initial LR check is under a threshold (exhaustive testing showed that this should be set to 20%), then an error flag is generated.

Finally, the third detection ($Fail3Det$) is done after the optimization step. This step checks if an error of the third type has occurred (parts of the image have been corrupted). The check is done similarly with the second one (using the left-right check over the cost volume), but we use a different error threshold (set to 75%).

If none of the errors have occurred, we perform the selection and refinement according to the main stereo method. Otherwise, we activate the error mitigation branch by activating the CNN.

IV. Stereo Failure Mitigation

A. Semantic information as reliable information

In order to increase the reliability by using object infromation, we initially compute a semantic segmentation of the scene. Classic segmentation methods have been used in correlation with depth computation [36] [37], generally being used as a stereo post-processing [38]. With the introduction of deep neural networks a boost in semantic segmentation has lately appeard. Cityscapes dataset [39] enables methods such as [40], [41] or [42] to accurately generate semantic labels at pixel level.

One of the top approaches in semantic segmentation is ErfNET [43]. The method uses an encoder-decoder architecture, with 23 layer blocks, providing one of the best trade-offs in terms of accuracy (69.7 for IoU) vs speed (around 25 ms). The robustness is especially given by their novel layer block that uses residual connections and factorized convolutions for preserving the structure in the image and also reducing computational costs. The method classifies the scene into 19 foreground and background classes.

B. CNN Architecture for disparity restoration

For the learning-based disparity generation we employ a 4-input ConvNet architecture (Fig. 3). The method follows the principles of the DeepJoint filter proposed by [7], a method initially introduced in the context of image upsampling.

Our CNN architecture consists in several sub-networks, each being responsible for solving a particular sub-problem. Each sub-network consists of three residual Non-bottleneck1D blocks, followed by a Batch Normalization layer. The first block contains 64 feature maps, the second 128, while the third produces just one feature map, that incorporates the most relevant features extracted from each branch. Each convolution layer is designed according to the speed-up techniques presented in [43]. A Non-Bottleneck1D (Fig. 4) block is therefore shaped by:

- Residual connections – important information extracted from initial layers is preserved throughout the entire network so that later layers can benefit from it;
- 2D convolution layers approximated by two 1D convolutions – this trick reduces the number of convolution weights by more than a half while preserving stability and accuracy;
- ReLU (Rectified Linear Unit) activations are inserted after each convolution; these are used to zero the gradients of negative input values.

The first two sub-networks are have similar roles, each of them extracting reliable information from previous and current left frame. This part has the role of extracting reliable features from two temporally successive RGB images from left camera. Two 41x41 patches from the are the inputs to these branches. The output of these branches are concatenated to the following sub-network, which has the role of extracting possible temporal differences between these successive frames.

Besides this temporal information extraction we add a sub-network that extracts relevant depth information. The input of this sub-network comes from the last reliable disparity (given by the last frame with information available from right image).

Exhaustive testing showed us that RGB features from current left image can not provide effective guidance. This problem is mainly caused by the mixture of information RGB maps carry. Although first sub-network tends to extract more effective features and provides relevant information, we consider useful to aid the process with an additional term. Therefore results of the temporal and depth branches are also concatenated with a semantic patch extracted from the segmentation image. The semantic segmentation map contains information about object boundaries, linking together similar structures.

The last part of the network consists in two additional Non-Bottleneck-1D blocks, interleaved by a layer of 1x1 convolutions. The role of the final sub-network is to simulate a non-linear regression, joining together the three maps thus inferring the final depth image. In this way we will integrate the knowledge extracted from the three aforementioned feature maps. In terms of loss function, a pixel-wise mean squared error (MSE, or L2) is computed between the resulting depth patch and a ground truth, thus estimating the degree of convergence for our method. Other losses could have been used for this problem (such as L1 or the semi-supervised loss described in [44]), but our network seemed to converge fine with MSE, without any additional overhead.

C. Parameters and Training Details

Extracted training patches are normalized by subtracting the mean and dividing with the maximum image intensity. Similar learning rates have been given to all input branches. Experimental testing showed that our network converged only when the segmentation learning rate was set to 1/5 of the learning rate for RGB, so we modified it accordingly. In other scenarios segmentation features became too powerful, and other information was dropped. We experimented with two optimization methods: Stochastic Gradient Descent and Adaptive Moment Estimation (Adam). Adam seemed to properly control the learning so we chose it as our optimizer.

The network has been trained for 400 epochs, with batch sizes of 128, the learning rate being decreased with a factor of 0.1 at each 100 epochs.

V. Evaluation

A. Dataset generation

Since we use supervised machine learning techniques for optimization, a reliable dataset is required. The main prerequisites for our training set are:

1) RGB images for semantic segmentation with driving scenarios
2) depth image acquired either from stereo (need left and right images)
3) left and right images captured in two consecutive frames, for temporal information and ground truth generation

Kitti 2015 stereo dataset [45] is the the most adequate choice for our task, meeting all our needs: it contains both left and right images in two consecutive frames thus making us able to generate images for real-life driving scenarios. The second pair of images are used for the ground truth disparity computation. For CNN input, we extract 41×41 patches at various positions and randomly shuffled such that we generate around 60.000 patches from 90% of the training subset, with RGB, semantic information, left and right images for frame t (for disparity) and left images for frame $t + 1$. These images have been further separated into

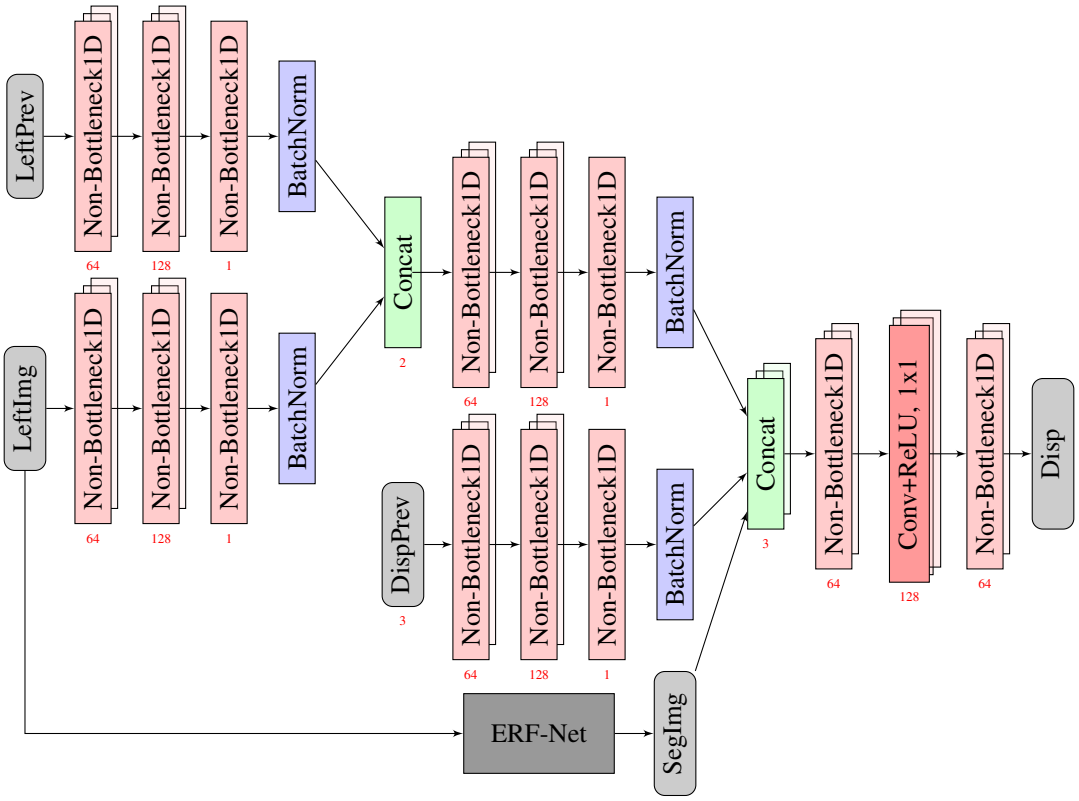


Fig. 3: Architecture of the Proposed Disparity Generation Method

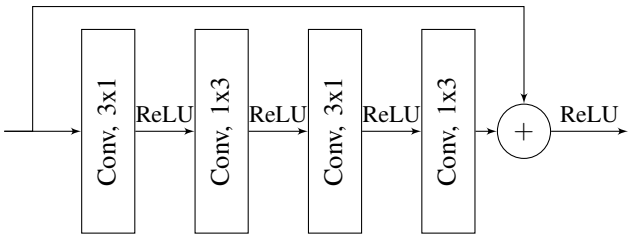


Fig. 4: Non-bottleneck1D block having a receptive field size of 5x5

training (80%) and validation (20%). The CNN is trained by using the Torch7 framework, on a Nvidia GTX 1080 GPU. In terms of segmentation, ERF-Net has been intially trained on the Cityscapes dataset [39]. Semantic patches are then obtained by normalizing the results according to the mean and standard deviation of Kitti images.

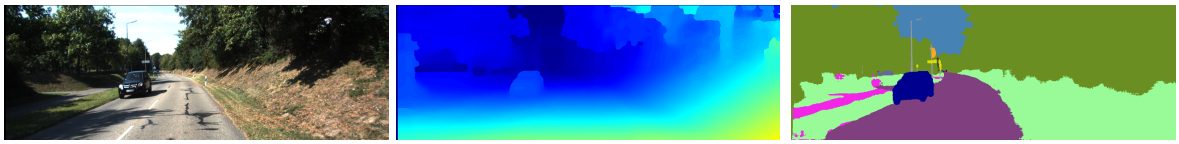
B. Accuracy of Disparity recovery

1) Method Accuracy: In order to properly test the efficiency of our CNN-based disparity recovery, we test the results obtained with our architecture with respect to several other counterparts. Since to the best of our knowledge there is no benchmark for this task (or any publicly available implementations that have similar objectives), we stress our CNN architecture such that it can be properly compared with the following methods:

- The CNN without the LeftPrev sub-network. Therefore, no temporal information is extracted in this case;
- The CNN without any prior disparity information. In this case there is no information about the last depth structure
- The CNN without semantic information. There is no information about boundaries, nor about object types.

We have trained all these methods for multiple epochs, using the same loss (MSE) as in our initial architecture. The error metric for evaluation is the number of mismatched pixels with respect to the ground truth (disparity given by a regular stereo), with a threshold error of 3 pixels. We used the other 20% of Kitti images, that have not been considered for training or evaluation.

A first observation is that the most important information for the architecture is the previous disparity map. Without this prior, the CNN behaves poorly. Both temporal and semantic information add an additional knowledge, the proposed architecture obtaining an overall error of 6.42% with respect to the original disparity. However, part of this error is caused by the stereo method itself, since the disparity also suffers from errors that can mislead the CNN. Numerical results are presented in Table II. Visual results can be seen in Figure 5. In this figure are presented all images that are contributing to the proposed pipeline: left current and previous image (Fig. 5a and 5d), disparity of the previous image 5b, semantic segmentation of the current image 5c and also the ground truth consisting in the disparity image without faults 5e. The disparity that has been inferred by using our method is presented in 5f.



(a) Left – Previous Image (b) Disparity – Previous Image (c) Semantic Segmentation – Current Image

(d) Left – Current Image (e) Ground Truth Disparity Image (f) Inferred Disparity Image using proposed solution

Fig. 5: Disparity maps obtained with our concealment method

TABLE II: Performance of various correction architectures for Kitti stereo images

Method	Accuracy	Speed
Without temporal info	11.36%	11 ms
Without old disparity	19.03%	14 ms
Without semantic info	9.74%	15 ms
Proposed architecture	6.42%	17 ms

TABLE III: Accuracy of the disparity when changing the base stereo method

Method	Error (regular)	Error (concealed)
OCV-BM	30.89%	35.12%
OCV-SGBM	14.75%	19.23%
MC-CNN fast	3.79%	7.17%
MC-CNN acrt	3.03%	6.64%
Proposed	10.19%	15.22%

2) Stereo method variation: We are also interested to see how our CNN method behaves when the underlying stereo method is changed. We chose the following stereo methods as possible variations for the basic stereo pipeline: Block Matching (BM) and Semi-Global Block Matching (SGBM) from OpenCV, MC-CNN fast, MC-CNN accurate from [5] and our method with Census+Aggregation+SGM+Median filter. For this evaluation we compare the stereo results with respect to the regular stereo Kitti ground truth (LiDAR points, accumulated from multiple frames).

The results obtained in this case can be seen in Table III. It can be seen that all these methods can be used as underlying stereo for our method, the error constantly increasing, regardless of the type of cost computation/optimization method used in stereo. Nevertheless, very accurate stereo methods have a smaller error increase (about 2%), mostly because all information passed to the CNN is consistent (eg. semantic with depth), and thus the CNN can better understand the image structure. All in all, our concealment CNN-based method can be used with any type of stereo.

3) Error increase with time: Although the Kitti stereo dataset provides a set of two consecutive stereo pairs, one of them containing also a ground truth, its images are randomly selected, and thus it can not provide a larger sequence of frames. In order to properly test the behavior of our method,

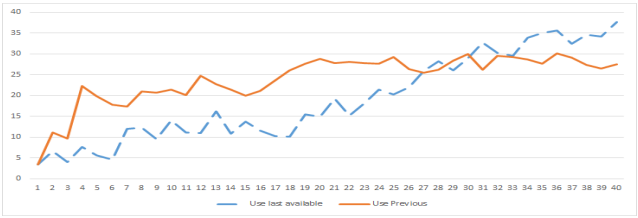


Fig. 6: Error for Kitti raw image sequence (40 frames)

we run our solution on a sequence from the raw Kitti dataset. The sequence consists in 40 consecutive traffic images. For this sequence we evaluate two types of approaches when using the aforementioned correction method. As input to our method we use as last available depth information:

- from the last disparity obtained with a regular stereo (that also had reliable information from right camera)
- from last available disparity (previous frame)

Results are presented in Figure 6, which depicts the error rate of the inferred disparity with respect to the frame number, starting from the last available frame. It can be seen that using the initial disparity (that consists in more reliable results) is useful for the initial (around 30) frames. After this point the information provided becomes too outdated, so using the last available frame (although more unreliable) leads to better results. The performance does not decreases drastically for several frames, because of the similar structure that all traffic images have. Nevertheless, these results also show that our method works only for short-term failures, since the error after 20 missing frames becomes too large ($> 20\%$).

VI. CONCLUSIONS

Convolutional neural networks are becoming more and more popular in depth measurement, their capabilities being extremely useful for autonomous vehicles perception. We have shown here a novel way in which such learning methods can be used – to conceal some of the possible camera errors that may occur in stereo reconstruction. We consider that such approaches are required in order to make stereo reconstruction more reliable and trustworthy for autonomous

cars. We intend to continue our work by developing other convolutional architectures meant to solve other depth perception problems such as upsampling or single image-based depth generation.

ACKNOWLEDGEMENT

This work was supported by UEFISCDI (Romanian National Research Agency) in the national research projects PN III PCCF SEPCA (Semantic Visual Perception and Integrated Control for Autonomous Systems), project no 9/2018 and Multispectral Environment Perception by Fusion of 2D and 3D Sensorial Data from the Visible and Infrared Spectrum (MULTISPECT), project code PN-III-P4-ID-PCE-2016-0727.

REFERENCES

[1] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 2, pp. 328–341, Feb 2008.

[2] R. Spangenberg, T. Langner, S. Adfeldt, and R. Rojas, "Large scale semi-global matching on the cpu," in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, June 2014, pp. 195–201.

[3] I. Haller and S. Nedevschi, "GPU optimization of the SGM stereo algorithm," in *Intelligent Computer Communication and Processing (ICCP), 2010 IEEE International Conference on*, Aug 2010, pp. 197–202.

[4] C. Banz, S. Hesselbarth, H. Flatt, H. Blume, and P. Pirsch, "Real-time stereo vision system using semi-global matching disparity estimation: Architecture and fpga-implementation," in *International Conference on Embedded Computer Systems (SAMOS)*, 08 2010, pp. 93 – 101.

[5] J. Zbontar and Y. LeCun, "Computing the stereo matching cost with a convolutional neural network," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 1592–1599.

[6] A. Seki and M. Pollefeys, "SGM-Nets: Semi-Global Matching With Neural Networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[7] Y. Li, J.-B. Huang, N. Ahuja, and M.-H. Yang, *Deep Joint Image Filtering*. Springer International Publishing, 2016, pp. 154–169.

[8] N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4040–4048, 2016.

[9] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry, "End-to-End Learning of Geometry and Context for Deep Stereo Regression," *CoRR*, vol. abs/1703.04309, 2017. [Online]. Available: http://arxiv.org/abs/1703.04309

[10] N. Schneider, L. Schneider, P. Pinggera, U. Franke, M. Pollefeys, and C. Stiller, "Semantically guided depth upsampling," in *GCPR*, 2016.

[11] J. Dolson, J. Baek, C. Plagemann, and S. Thrun, "Upsampling range data in dynamic environments," *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1141–1148, 2010.

[12] F. Liu, C. Shen, G. Lin, and I. Reid, "Learning depth from single monocular images using deep convolutional neural fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2024–2039, Oct. 2016. [Online]. Available: https://doi.org/10.1109/TPAMI.2015.2505283

[13] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[14] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, vol. 47, no. 1, p. 742, May 2002. [Online]. Available: http://research.microsoft.com/apps/pubs/default.aspx?id=64200

[15] W. van der Mark and D. M. Gavrila, "Real-time dense stereo for intelligent vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 1, pp. 38–50, March 2006.

[16] R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," in *Computer Vision ECCV '94*, ser. Lecture Notes in Computer Science, J.-O. Eklundh, Ed. Springer Berlin Heidelberg, 1994, vol. 801, pp. 151–158.

[17] E. Tola, V. Lepetit, and P. Fua, "DAISY: An Efficient Dense Descriptor Applied to Wide-Baseline Stereo," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 815–830, May 2010. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2009.77

[18] Y. S. Heo, K. M. Lee, and S. U. Lee, "Mutual information-based stereo matching combined with SIFT descriptor in log-chromaticity color space," *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 445–452, 2009.

[19] W. Luo, A. G. Schwing, and R. Urtasun, "Efficient deep learning for stereo matching," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 5695–5703.

[20] Z. Chen, X. Sun, L. Wang, Y. Yu, and C. Huang, "A deep visual correspondence embedding model for stereo matching costs," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[21] A. Kuzmin, D. Mikushin, and V. S. Lempitsky, "End-to-end Learning of Cost-Volume Aggregation for Real-time Dense Stereo," *CoRR*, vol. abs/1611.05689, 2016. [Online]. Available: http://arxiv.org/abs/1611.05689

[22] V. Kolmogorov and R. Zabih, "Computing visual correspondence with occlusions via graph cuts," Ithaca, NY, USA, Tech. Rep., 2001.

[23] J. Sun, N.-N. Zheng, and H.-Y. Shum, "Stereo matching using belief propagation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 7, pp. 787–800, Jul. 2003. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2003.1206509

[24] T. Huang, G. Yang, and G. Tang, "A fast two-dimensional median filtering algorithm," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 27, no. 1, pp. 13–18, Feb. 1979.

[25] C. Tomasi and R. Manduchi, "Bilateral Filtering for Gray and Color Images," in *Proceedings of the Sixth International Conference on Computer Vision*, ser. ICCV '98. Washington, DC, USA: IEEE Computer Society, 1998, pp. 839–. [Online]. Available: http://dl.acm.org/citation.cfm?id=938978.939190

[26] K. He, J. Sun, and X. Tang, "Guided Image Filtering," in *Proceedings of the 11th European Conference on Computer Vision: Part I*, ser. ECCV'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 1–14. [Online]. Available: http://dl.acm.org/citation.cfm?id=1886063.1886065

[27] Y. Chen, C. Cai, and K.-K. Ma, "Stereoscopic video error concealment for missing frame recovery using disparity-based frame difference projection," in *2009 16th IEEE International Conference on Image Processing (ICIP)*, Nov 2009, pp. 4289–4292.

[28] T. Y. Chung, S. Sull, and C. S. Kim, "Frame loss concealment for stereoscopic video plus depth sequences," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 3, pp. 1336–1344, August 2011.

[29] T. L. Lin, T. E. Chang, G. S. Huang, and C. C. Chou, "Multiview video error concealment with improved pixel estimation and illumination compensation," in *2013 International Symposium on Intelligent Signal Processing and Communication Systems*, Nov 2013, pp. 157–162.

[30] M. Yang, X. Lan, N. Zheng, and P. Cosman, "Depth-assisted temporal error concealment for intra frame slices in 3-d video," *IEEE Transactions on Broadcasting*, vol. 60, no. 2, pp. 385–393, June 2014.

[31] M. Ranjbari, A. Sali, H. A. Karim, and F. Hashim, "Depth error concealment based on decision making," in *2013 IEEE International Conference on Signal and Image Processing Applications*, Oct 2013, pp. 193–196.

[32] J. Fridrich, "Sensor defects in digital image forensic," pp. 179–218, 11 2013.

[33] T. Sanislav, G. Mois, and L. Miclea, "An approach to model dependability of cyber-physical systems," *Microprocess. Microsyst.*, vol. 41, no. C, pp. 67–76, Mar. 2016. [Online]. Available: http://dx.doi.org/10.1016/j.micpro.2015.11.021

[34] "Analysis Techniques for System Reliability - Procedure for Failure Mode and Effects Analysis (FMEA)," International Standard IEC 60812, Tech. Rep., 2006.

[35] X. Mei, X. Sun, M. Zhou, S. Jiao, H. Wang, and X. Zhang, "On building an accurate stereo matching system on graphics hardware," in *ICCV Workshops*. IEEE, 2011, pp. 467–474.

[36] K. Yamaguchi, D. McAllester, and R. Urtasun, "Efficient joint segmentation, occlusion labeling, stereo and flow estimation," in *European Conference on Computer Vision*. Springer, 2014, pp. 756–771.

[37] V.-C. Miclea and S. Nedevschi, "Semantic segmentation-based stereo reconstruction with statistically improved long range accuracy," in *Intelligent Vehicles Symposium Proceedings, 2017 IEEE*, 06 2017, pp. 1795–1802.

[38] M. Humenberger, T. Engelke, and W. Kubinger, "A census-based stereo vision algorithm using modified Semi-Global Matching and plane fitting to improve matching quality," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, June 2010, pp. 77–84.

[39] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[40] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 3431–3440.

[41] G. Ghiasi and C. C. Fowlkes, "Laplacian reconstruction and refinement for semantic segmentation," *CoRR*, vol. abs/1605.02264, 2016. [Online]. Available: http://arxiv.org/abs/1605.02264

[42] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "Enet: A deep neural network architecture for real-time semantic segmentation," *CoRR*, vol. abs/1606.02147, 2016. [Online]. Available: http://arxiv.org/abs/1606.02147

[43] E. Romera, J. M. lvarez, L. M. Bergasa, and R. Arroyo, "Erfnet: Efficient residual factorized convnet for real-time semantic segmentation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 263–272, Jan 2018.

[44] C. Godard, O. Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," 09 2016.

[45] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.