Fault Tolerant 3D Reconstruction using Fusion of Complementary Depth Estimation Approaches

Mircea Paul Muresan
Computer Science Department
Technical University of Cluj-Napoca
Cluj-Napoca, Romania Mircea.Muresan@cs.utcluj.ro

Sergiu Nedevschi
Computer Science Department
Technical University of Cluj-Napoca
Cluj-Napoca, Romania Sergiu.Nedevschi@cs.utcluj.ro

***Abstract*—Accurate depth estimation from cameras is a highly important task for autonomous systems. Existing monocular depth estimation methods span a wide range of approaches, from supervised techniques which leverage labelled datasets, to self-supervised methods which utilize video sequences without explicit depth annotations. Additionally, stereo vision solutions provide absolute depth measurements using the disparity maps obtained via stereo correspondence. This paper proposes a depth fusion system combining Monodepth2, MiDaS and stereo vision using semantic-aware scaling and error-aware selection. The original contributions of our work are three-fold. First, our approach proposes an original method of scaling the relative disparities from monocular depth estimation using semantic segmentation and an original depth discretization technique. Then, an original method is proposed for combining self-supervised and supervised approaches using the probability of information obtained from learning the error each type of monocular depth estimation method produces. The final contribution consists in the presentation of the fault tolerant system used to reconstruct the scene. The proposed approach has been tested on the KITTI dataset and highlights the effectiveness of combining these complementary methods showing good results even in situations where individual methods would fail.**

Keywords— Depth Estimation, Monocular Depth, Stereo Vision, Information Fusion, Multimodal Depth Estimation

I. INTRODUCTION

Depth estimation is a fundamental task in environment perception for autonomous systems because it tries to recognize the spatial structure of the scene where the robot must safely navigate. A wide range of sensors can be used to extract depth information and among the most popular in the autonomous driving filed are LiDARs, Radars and cameras. Each of these sensors have their advantages and disadvantages when working in real world scenarios, and typically for a real world autonomous system to navigate safely the redundant sensorial information is fused [1]. Cameras have attracted the attention of researchers due to their ability to obtain semantic information of the environment, they do not have moving mechanical parts and their price is relatively cheaper compared to other sensors.

To accurately infer the depth of a scene from a 2D image multiple challenges must be overcome such as varying lighting conditions, occlusions, or the inherent ambiguities of depth cues in monocular images.

This work is supported by the project "Romanian Hub for Artificial Intelligence - HRIA", SMIS no. 351416.

Through the years, numerous depth estimation solutions [2,3] have been developed each with its strengths and weaknesses. Broadly, based on the number of cameras, the most popular depth estimation methods used in autonomous driving can be classified into monocular depth estimation methods, when only one camera is available, and stereo reconstruction methods, when two cameras are available. The monocular depth estimation methods can be further classified into supervised and self-supervised methods depending on the approach used to train the deep neural network that outputs the depth information.

Supervised approaches [4] rely on large datasets with ground truth annotations to train the neural networks to infer the depth from a single image. While the results are impressive, obtaining large, labelled datasets is an expensive and time-consuming task. Self-supervised methods [5] have emerged to address the issue of data dependency by training monocular depth estimation methods on single image video sequences without requiring ground truth information. These methods, rely on photometric consistency between adjacent frames to infer depth, however they struggle with scale ambiguity and can be less accurate in static scenes.

When monocular depth estimation models are not trained on annotated datasets with metric depth values — or when the training setup does not enforce metric consistency — the resulting depth maps are usually scaled relatively rather than providing absolute measurements. This is due to the inherent limitations of monocular images that only provide depth cues without direct information about the actual distances in the scene. The relative depth is converted in absolute measurement using a scaling factor [6,7] that adjusts the depth values from the monocular algorithm to real world distances. Finding an accurate scaling factor is a difficult task and a crucial endeavour for applications requiring precise depth measurements. The incorrect scaling of the depth can lead to inaccuracies in the scene reconstruction and can have disastrous effects on other modules of an autonomous system.

Stereo vision sensors [8] can offer dense depth maps that are particularly effective when providing absolute depth measurements which are very important to applications where scale consistency is important. Despite their accuracy stereo methods can be affected by multiple issues such as low texture regions, repetitive surfaces, occlusions or the inherent assumption that the distances within matching windows are the same which is false on slanted surfaces or due to the perspective effect. All these issues can lead to the deterioration of the depth map. Moreover, if for any reason one of the cameras of the stereo system is not providing

979-8-3315-5896-3/25/$31.00 ©2025 IEEE

images the stereo reconstruction cannot happen and in an urban environment where a robot must navigate safely this can have unwanted outcomes.

Given the complementary nature of these types of reconstruction methods, there is significant potential in combining them to leverage their advantages while eliminating most of their shortcomings. In this paper, a novel framework is proposed that integrates supervised, self-supervised, and stereo depth estimation techniques to create a more robust, fault-tolerant, and accurate depth prediction system. By fusing the strengths of different reconstruction solutions, our approach addresses the limitations of individual methods, resulting in improved performance across a variety of environments and conditions.

Based on the aforementioned motivation, this paper contributes the following:

- A novel scaling mechanism that uses semantic information and depth discretization to transform the relative depth estimation in absolute measurements. This approach also leverages an adapted online pre-scaling method called Semantic Dense Geometrical Constraint (SDGC) that is used for an initial relative depth scaling.
- A fusion mechanism between different types of monocular depth estimation approaches by using the probability map resulted from learning the error each type of method produces.
- A fault tolerant approach that combines the stereo and monocular depth estimation methods such that a depth map can be always retrieved even when one of the cameras is not working properly

The proposed approach has been evaluated on the KITTI [9] dataset, and the results clearly demonstrate increased accuracy when combining the depth estimation methods as opposed to individual approaches. Moreover, the results indicate that by combining different depth estimation paradigms, a more reliable depth estimation system can be achieved even when one of the sensors is not operational.

II. Related Work

Metric depth estimation (MDE) methods that were originally implemented, leveraged labelled datasets and used supervised learning approaches to train the deep learning models. Notable here the work of Eigen et al.[10] proposed a continuous regression multi-scale CNN that combined global and local features for improving depth reconstruction accuracy. Following this approach the authors boosted the performance of their original method by using surface normal and semantic segmentation via a multi-task network[11].

A broad diversification of the methods used to estimate depth followed in recent years. For example, a domain shift occurs where the methods transition from continuous regression to discrete modelling techniques. Newer methods strategies discretize depth into intervals and frame the depth estimation task as a classification problem [12], rather than regressing raw depth values. Fu et al. [13] build on this idea and proposed a regression model that applies a non-linear depth binning scheme and an ordinal-aware loss function to further evolve the initial idea. This approach laid the foundation for later works [14]-[18], which created hybrid models combining regression and classification, and dynamically adapting binning. Zoedepth [18], created an automatic routing mechanism that selects the appropriate prediction head based on whether a scene is classified as indoor or outdoor.

An alternative to metric depth estimation is relative depth estimation (RDE), which, rather than predicting absolute depth values, aims to estimate depth orderings between image regions. More specifically, RDE solutions focus on determining if one point is closer or farther away than another [19], or on determining the relative spatial arrangement of scene elements [20]. This approach has the advantage of being robust to changes in scale, as the ordinal depth relations remain consistent regardless of camera internal parameters, or the actual range of the scene. The authors from [19] and [21] employed CNNs to predict the relative depth between pixel pairs, while also considering properties specific to real world scenes such as reflectance and illumination. Using these examples as a theoretical foundation, subsequent studies have explored ways of combining RDE and MDE pipelines to improve generalization as well as to reduce the need of metric annotations. In this way Chen et al. [22] demonstrated that some models can achieve competitive metric predictions in complex uncontrolled environments even when trained with sparse relative depth labels. The authors from [24] introduced a scale invariant loss function to mitigate the scale ambiguity issues, and that was later refined in subsequent works such as [23] and [25], enabling models to make consistent predictions across varying scale ranges.

Jun et. al. proposed in [26] a decomposition strategy based on two branches in a deep net, that run in parallel, and which separately estimate relative and metric depth. Recent models such as Depth Anything [27] and Marigold [28] have demonstrated amazing generalization across diverse datasets. However, even these approaches output relative depth maps and require additional scaling transformations to recover metric representations, not to mention the dimensions of the models which are very large. Some approaches from the literature such as [29] and [30] make a fail-safe mechanism for real time depth estimation methods, toggling between stereo and monocular depth perception. These approaches ensure that if only one image is captured some depth estimation is offered to the downstream perception tasks.

III. Proposed Solution

The Proposed Solution section is divided into three subsections. The first subsection describes the novel approach used to transform relative depth information into absolute measurements using semantic data. The second subsection presents the strategy employed to fuse the complementary monocular depth estimation modules, and the final subsection outlines the fault-tolerant architecture of the proposed system. In this work, MonodepthV2 is employed as the self-supervised monocular depth estimation model, MIDAS for the supervised monocular depth estimation model and a stereo approach developed by us based on the paper of Hirschmuler [31]. Although MiDaS is trained in a supervised manner using multiple labeled datasets, its predictions are not in metric scale and require external scaling to be used as absolute depth. Therefore, both monocular approaches are scaled using the method described in Section III.A before fusion. It is worth noting that the proposed solution can work with any type of depth estimation algorithm and is not limited by the methods mentioned above.

A. Semantic Aware Scaling

Monocular depth estimation methods only estimate a relative relation between depths but cannot provide the absolute distance without additional information. Scaling is necessary to adjust the depth estimate to a real scale consistent with the physical world. In the proposed solution, a semantic segmentation architecture, DeepLabV3+ [32], trained on the KITTI dataset, is used to extract semantic information from the scene.

At runtime two scenarios emerge. In the first scenario the images from both cameras are available. The stereo reconstruction will happen and the resulted depth map is used as reference to compute the scaling factor. In the second scenario, only one image is available, due to one of the images from one of the two cameras being corrupted or missing. In this scenario there are two main steps an online step (that is semantic class agnostic) and an offline step (that is computed only once in a pre-processing step). For the preprocessing offline step the KITTI 2015 stereo data flow LiDAR ground truth is used as reference and the general steps of the procedure are the same as in the case when the depth image is computed using the two available images (only this time is computed once using the LiDAR ground truth).

The data used for scaling, depending on the scenario, will be referred to as the reference image.

The first scenario considers the case in which both images are available, allowing the computation of the stereo depth map through stereo reconstruction.

The points from each monocular depth estimation method (MonodepthV2 and MIDAS) and from the reference image are transformed from disparity space to depth space. The number of different semantic classes is extracted from the semantic segmentation image. For every semantic class, all the 3D points from each monocular depth estimation solution and from the reference image, corresponding to that semantic class are extracted and stored in a separate vector. Therefore, for each semantic class for every monocular depth estimation method as well as for the reference data there will be a separate vector. All vectors corresponding to each semantic class are sorted in ascending order. A discretization of the depth is then considered, where the vectors are divided into T bins containing an equal number of points each. Next, for each bin, the median value is chosen. A ratio is performed to find the scaling factor between the median value of each bin and the median value from the reference image from the corresponding bin, for each semantic class. This operation is performed separately for each monocular depth estimation solution. The value of T has been found empirically and has the value of 400 bins in our solution. Additionally for each bin the minimum and maximum values of depth are also computed.

After obtaining the scaling factor for each bin of each semantic class, the scaling factors are applied. The unscaled depth image is traversed, and the semantic class corresponding to the coordinates of each unscaled point is extracted. From the vector associated with that class, the bin in which the unscaled depth value is located is determined using the previously obtained minimum and maximum values. The scaling factor for the corresponding interval is then applied, and the resulting scaled point is stored. To mitigate effects such as banding or other artifacts that may arise when using interval-based scaling algorithms, linear interpolation is employed between scaling factors to ensure a smooth transition. The interpolation is done according to equations (1) and (2) below.

$$t = \frac{depth - minVals_{classIdx}[k]}{maxVals_{classIdx}[k+1] - minVals_{classIdx}[k]} \quad (1)$$

$$scale = scaleFactors_{classIdx}[k](1-t) + scaleFactors_{classIdx}[k+1]t \quad (2)$$

In equation (1) the meaning of the terms are the following: depth represents the depth value of the current pixel being evaluated, minVals is a vector that contains the minimum depth values for each semantic class and for each interval within that class (minValues[classIdx][k] represents the minimum depth value for the k-th interval of the classIdx class), maxVals is a vector that contains the maximum depth values for each semantic class and for each interval within that class, t represents the relative position of the depth value within the identified interval, measured from the lower bound (minValues[k]) to the upper bound (maxValues[k+1]); t will be a number between 0 and 1. In equation (2) the meaning of the terms is the following: scaleFactors represents a vector that contains the scaling factors for each semantic class and for each interval within that class (scaleFactors[classIdx][k] is the scaling factor associated with the lower bound of the k-th interval for the classIdx class and scaleFactors[classIdx][k + 1] is the scaling factor associated with the upper bound of the (k+1)-th interval for the classIdx class), scale represents the scaling factor calculated for the depth value depth (it is obtained by linearly interpolating between the two scaling factors specific to the interval where depth falls).

In the absence of the stereo depth image, due to failure of one of the cameras, the procedure consists of offline and online steps. For this scenario, a general scaling factor is computed for each frame using the Semantic Dense Geometrical Constraint (SDGC) method for the entire image. Subsequently, scaling factors for each semantic class are computed as previously described. In this approach, the scaling factors and interval limits for each bin of each semantic class are computed offline as averages, using LiDAR ground truth as a reference instead of the stereo depth image. The averaging is performed over all KITTI 2015 training images. During runtime, the initial scale is computed using SDGC, followed by the application of the individual class-specific scaling factors as described previously.

The method proposed for scaling the entire image is referred to as Semantic Dense Geometrical Constraint (SDGC) and constitutes an original adaptation of DGC [33] that additionally incorporates semantic information. For completeness, the entire process employed by SDGC is overviewed. First, the relative 3D points are obtained, followed by estimation of the surface normal for each point. From the relative depth map, the 3D points are reconstructed, and the surface normal is estimated in the vicinity of each point. An 8-neighbourhood around each point is considered, and four planes are created for which the surface normals are computed. The final normal is determined as the average of these normals, as illustrated in equation (3).

$$N(P_{i,j}) = \frac{1}{4}\sum_{g=1}^{4}\frac{n_g}{||n_g||} \quad (3)$$

Subsequently, ground points are identified. A pixel is classified as a ground point if its surface normal is close to the ideal ground normal and the semantic class of the pixel corresponds to the ground type. This process is illustrated in equations (4) and (5)

$$s(P_{i,j}) = \arccos\left(\frac{\tilde{n} * N(P_{i,j})}{||\tilde{n}|| * ||N(P_{i,j})||}\right) \quad (4)$$

$$GM(i,j) = \begin{cases} 1, if\ s(P_{i,j}) < S_{max}\ and\ Semantic(i,j) = road \\ 0, otherwise \end{cases} \quad (5)$$

For each detected ground point, the estimated camera height is the projection of the 3D point onto its surface normal. This is illustrated analytically in equation (6) and intuitively in Figure 1.

$$H(P_{i_1,j_1}) = N(P_{i_1,j_1})^T * \overrightarrow{OP_{i_1,j_1}} \quad (6)$$

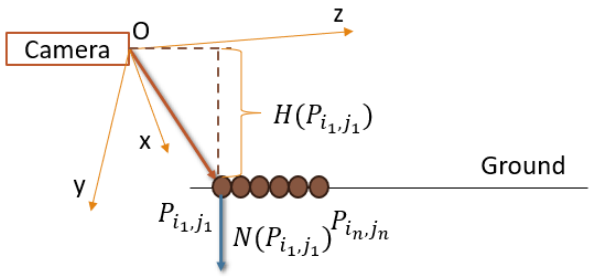

Figure 1. Intuitive depiction of the height estimation process

The final camera height is the median of all estimated heights (hM). Given the known real camera height (from the extrinsic calibration), the scale factor is computed as in (7)

$$ft = \frac{h_R}{h_M} \quad (7)$$

Finally, the scaled values are obtained by multiplying the relative 3D points by the scaling factor.

For the scenario where the stereo depth map is available the scaled fused monocular depth estimation results will be used to fill in regions that were not successfully reconstructed(for example in regions with repetitive structures or unstructured regions).

When the stereo image is available the scaling is performed at runtime using the method previously described and the monocular depth estimation values are used to fill unreconstructed regions of the stereo reconstruction.

B. Monocular Depth Fusion

For fusing the monocular depth estimation images, an innovative approach is employed in which data combination is performed using error probabilities from the semantic segmentation of the error learned for each point after the scaling operation. For this segmentation task, a DeepLabV3 architecture is utilized. Instead of relying on predictive uncertainty via ensembles or dropout, a lightweight segmentation-based estimation of depth error regions is applied using LiDAR-based ground truth, motivated by the need for real-time performance. The idea is to treat the error map as a binary semantic segmentation task and learn the likely error regions for each depth estimation method.

The dataset is first generated using the KITTI 2015 and 2012 ground truth data. For each monocular depth estimation model, the scaling procedure described in the previous section is applied, followed by computation of the error maps using the KITTI ground truth LiDAR data. The scaled values are compared with the ground truth depth values, and a pixel is considered erroneous if the ratio exceeds a predefined threshold; otherwise, it is considered correctly reconstructed. In this case, the threshold is set to 1.25. This is done for the left and right images to increase the size of the dataset. Moreover, the dataset is also augmented using a horizontal flip operation resulting in a total of 1600 images. We then generate the corresponding labels for each error image which we treat as a semantic image annotation and split the dataset in train, test and validation considering 70% for train, 20% for validation and 10% for test.

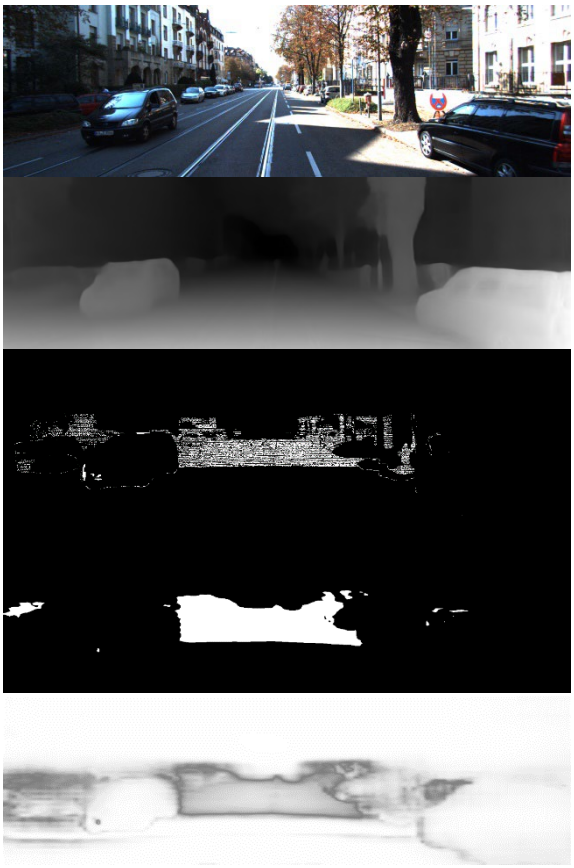

Figure 2. Result of training the semantic segmentation to identify the error map of a scene

The error segmentation model is trained for 350 epochs for each type of depth estimation model. The numerical results of the training are shown in the evaluation section.

After training a softmax is applied on the output of the network to convert the logits in a set of probabilities in the 0 and 1 interval.

Figure 2 shows the result of the segmentation on an unseen scenario. The top image represents the original RGB image, next is the disparity image obtained from Monodepth2 which has been scaled, then the error map computed using the KITTI ground truth LiDAR data, next is the result of the semantic segmentation which shows where the errors are detected, the final image show the probability map of the segmentation, the closer the colour value is to a darker tone the less probable the semantic class is. The same operation has been applied on the monocular supervised approach.
The probability maps are used to combine the data from the two monocular depth estimation models.

To obtain the final depth map, the probability maps are iterated, and values from the depth estimation method with the highest probability of correct detection are selected. The error maps of the individual depth estimation methods as well as their fusion are presented in Figure 3. The overall result is visibly improved, even though some of the errors that appear in both methods remain. The white regions represent

erroneous regions while the black ones are regions where the depth is correct. In Figure 3 the top image represents the error obtained from the Monodepth2 algorithm, following is the error probability for Monodepth2, the next image is the error map for the MIDAS algorithm and its error probability map, and finally the last image represents the error map of the fusion. The scene presented in Figure 3 is the same one as in Figure 2. The evaluation has been done on the KITTI dataset on images that were not used for training.

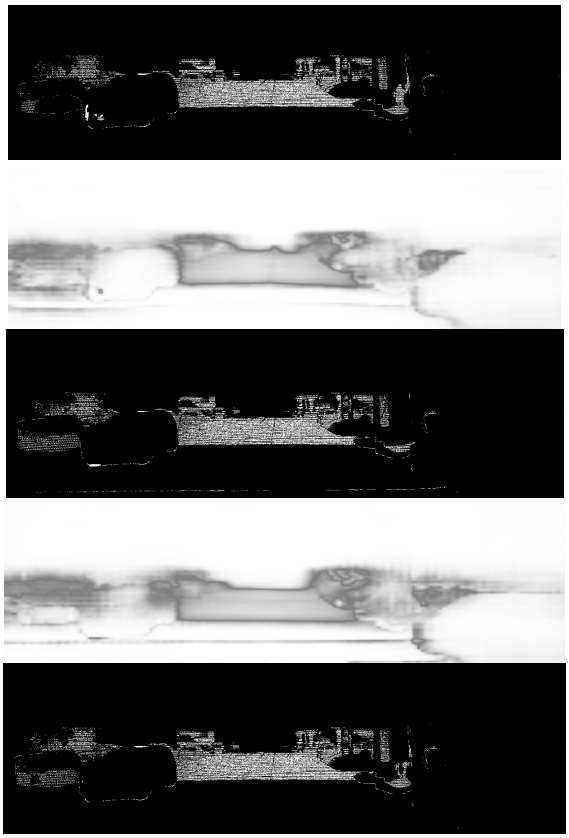

Figure 3. The error maps and the corresponding probability maps of Monodepth2 and Midas. The bottom image illustrates the fusion of the two methods.

C. *Fault Tolerant Depth Estimation Architecture*

The fault tolerant depth estimation architecture ensures that at each time moment the autonomous system will have a depth map on which it can rely to navigate safely in the environment. The dense stereo matching methods are more robust in terms of accuracy however they rely on two images to reconstruct the scene. Images can be corrupted for various reasons such as sensor issues or over saturation. When there is only one image available the system should be able to estimate the depth using that image. For evaluating image quality, a histogram is computed on the lower part of the image (the sky and upper part are not relevant for autonomous ground vehicles) using 10 randomly selected patches of 20×20 pixels each. If the mean intensity of the image is below a threshold T1 (set experimentally to 15) or above T2 (set to 245), the image is considered unusable. This test is performed on both images. If both images are usable the stereo vision algorithm is used, having as reference the left image, together with the monocular algorithms on the left image. The relative depth obtained through monocular depth estimation is scaled using the absolute depth from stereo as reference. The monocular depth estimation from the supervised and self-supervised method are combined using the methods presented in the previous section. The unreconstructed regions from the stereo are filled with the data from the fused monocular depth estimates. If only one image is available, the monocular depth estimation fusion presented in section 3 B is used. The system diagram of the fault tolerant system is presented in Figure 4.

The Frame 1 and Frame 2 blocks from Figure 4 represent the images coming from the two cameras, the Frame Consistency module checks if the two images are consistent using the algorithm presented before and provides to the Scale Supervised and Scale Self-Supervised modules the consistent image from the two frames. If both images are consistent then the left image is used. Moreover, the Frame Consistency module tells the system if it should use the monocular depth estimation, the stereo approach or just display a warning sign in case no frame is available to reconstruct the scene. The Scale Supervised and Scale Self-Supervised modules compute the monocular depth estimation and scale the relative depth maps using the algorithm presented in section 3 A. The disparity map from the stereo module is provided in cases both frames are consistent to scale the monocular depth estimation algorithms using this information. The stereo module computes the stereo disparity map if the two frames are consistent. The Mono Fusion block combines the two complementary monocular depth estimation methods, and the Depth Fusion module combines the stereo result with the fused monocular depth.

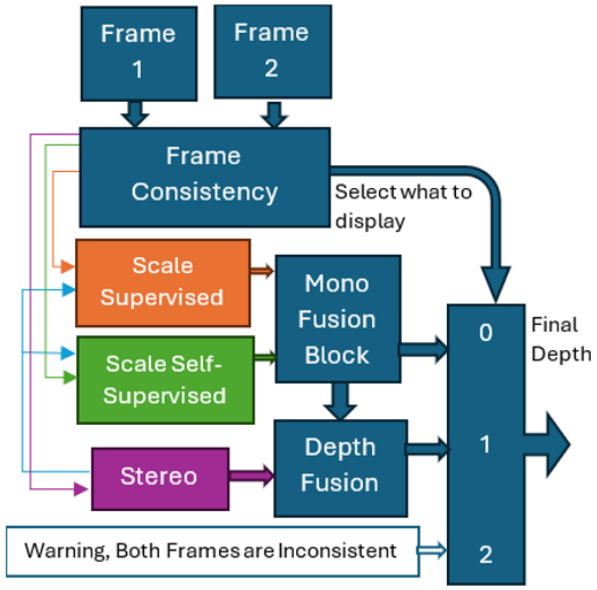

Figure 4. Block diagram of the fault tolerant system

IV. Experimental Results

For evaluating the proposed solution, the KITTI benchmark was used, which provides traffic images along with depth ground truth data acquired from LiDAR sensors. The specification of the system on which the solution was implemented has an 11th generation Intel processor i7-11370 running at 3.3 GHz, 4 Cores and 8 logical processors, 40.0 GB DDR4 memory and an Nvidia GForce RTX 3070 GPU. The networks used in this paper were trained using Pytorch and then traced in Libtorch to be used with C++. Other frameworks used for visualization purposes are point cloud library and OpenCV.

For training the DeepLabV3+ semantic model to segment regions potentially containing errors, the dataset was first generated and then split into 70% for training, 20% for validation, and 10% for testing. Evaluation on the test set yields a mIoU of 89.5 and an accuracy of 95.1% for the MIDAS CNN error map, and a mIoU of 86.7 and an accuracy of 93.5% for MonoDepth v2.

The running time of the whole solution is approximately 110ms using GPU and CPU optimizations. This includes both monocular depth estimation models running in parallel as well as semantic segmentation and stereo reconstruction.

The accuracy of the scaled fusion was computed using the KITTI 2015 dataset ground truth. For each image equation (8) was used, where TH has the values 1,2 and 3.

$$\left|\frac{GT_{depth}}{ObtainedDepth}\right| > TH \quad (8)$$

For the given threshold values the results obtained for the fused monocular depth estimation are illustrated in Table I. It is worth mentioning that the results have been averaged on the obtained values across the dataset and transformed the erroneous pixels result to percentage. Accuracy of the Monocular depth fusion on the KITTI Dataset

TABLE I. ACCURACY OF THE MONOCULAR DEPTH FUSION ON THE KITTI DATASET

Threshold	Nr of erroneous pixels
1	5.2877%
2	2.0955111%
3	0.0208149%

Table II illustrates the comparative performance of the fused monocular depth estimation and baseline methods on the KITTI dataset, using a threshold of 1.25 and a reduced image size. All methods were tested using the same resized KITTI images at a resolution of 832×256, chosen to balance runtime efficiency with spatial detail. For fairness, depth maps from all methods were scaled to a common metric scale using our semantic-aware scaling approach, when required.

TABLE II. COMPARISON OF THE PROPOSED MONOCULAR DEPTH FUSION METHOD WITH RESPECT TO THE BASELINE METHODS

Method	δ > 1.25 (Lower is better)
MIDAS	21.8
Monodepth	12.1
Fused	8.25

As can be seen from Table II, the fused approach offers overall better results than the individual methods. This aspect could also be seen visually from Figure 3.

Table III presents a comparison of the proposed fusion approach with other methods from the literature using the KITTI dataset. The proposed fusion approach proves better on the KITTI dataset than other approaches from the literature. In comparison to [18] the proposed solution is lightweight from the point of view of resource consumption. It is worth noting that the quality of the reconstruction depends on the size of the input image.

TABLE III. COMPARISON OF THE PROPOSED MONOCULAR DEPTH FUSION METHOD WITH OTHER METHODS

Method	δ < 1.25 (Higher is better)
PWA[36]	95.8
BTS[35]	95.6
AdaBins[34]	96.4
Fused	96.5
ZoeDepth[18]	96.8

With respect to some fault tolerant approaches from the literature such as the one presented in [29] the proposed approach also considers a fusion of different complementary monocular depth perception methods and thus can better adapt to unseen scenarios. In Figure 5, the result obtained after combining the monocular fused result with the stereo information.

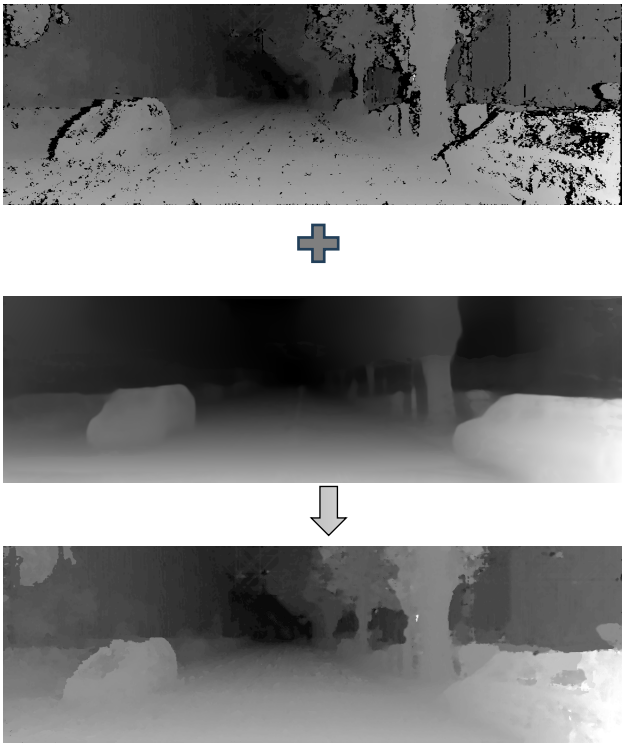

Figure 5. The combination between the fused complementary monocular approaches and the stereo reconstructed image.

In Figure 5, the top image illustrates the result obtained through stereo reconstruction, the middle image corresponds to the monocular fusion approach, and the bottom image depicts the integrated outcome of both methods. The black regions indicate areas that were not accurately reconstructed in the stereo process; these have been subsequently filled using information provided by the monocular fusion method.

CONCLUSIONS

This paper presents a 3D depth reconstruction system that fuses complementary depth estimations from MonoDepth2, MiDaS, and stereo vision using semantic-aware scaling and error-driven selection. Although each depth estimator performs differently across scene types and distances, their combination results in a more robust and consistent depth prediction.

The proposed system employs a novel scaling technique based on semantic segmentation and discretized ground truth statistics to convert relative monocular depth maps to a common scale. Additionally, an efficient method is introduced to detect potential errors in each depth map using a binary semantic segmentation network trained on binary error masks, allowing fusion of only the most reliable depth values at each pixel location. Furthermore, the approach includes an adapted version of the DGC method, named SDGC, and presents the architecture of a fault-tolerant depth estimation system that switches between different depth estimation methods based on the number of correctly acquired images.

The results on the KITTI dataset demonstrate improvements in individual methods and over several recent approaches. The architecture is computationally efficient and suitable for real-time applications

ACKNOWLEDGMENT

This work is supported by the project "Romanian Hub for Artificial Intelligence-HRIA", Smart Growth, Digitization and Financial Instruments Program, SMIS no. 351416.

REFERENCES

[1] M. P. Muresan, I. Giosan, and S. Nedevschi, "Stabilization and validation of 3D object position using multimodal sensor fusion and semantic segmentation," *Sensors*, vol. 20, no. 4, p. 1110, 2020.

[2] V. Arampatzakis, G. Pavlidis, N. Mitianoudis and N. Papamarkos, "Monocular Depth Estimation: A Thorough Review," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 4, pp. 2396-2414, April 2024, doi: 10.1109/TPAMI.2023.3330944.

[3] F. Rong, D. Xie, W. Zhu, H. Shang and L. Song, "A Survey of Multi View Stereo," *2021 International Conference on Networking Systems of AI (INSAI)*, Shanghai, China, 2021, pp. 129-135, doi: 10.1109/INSAI54028.2021.00033.

[4] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler and V. Koltun, "Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-Shot Cross-Dataset Transfer," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 3, pp. 1623-1637, 1 March 2022, doi: 10.1109/TPAMI.2020.3019967

[5] C. Godard, O. M. Aodha, M. Firman and G. Brostow, "Digging Into Self-Supervised Monocular Depth Estimation," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South), 2019, pp. 3827-3837, doi: 10.1109/ICCV.2019.00393.

[6] McCraith, R., Neumann, L., Vedaldi, A.: Calibrating self-supervised monocular depth estimation. arXiv preprint arXiv:2009.07714 (2020)

[7] K. Swami, A. Muduli, U. Gurram and P. Bajpai, "Do What You Can, With What You Have: Scale-aware and High Quality Monocular Depth Estimation Without Real World Labels," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, New Orleans, LA, USA, 2022, pp. 987-996, doi: 10.1109/CVPRW56347.2022.00112

[8] Muresan, M.P., Nedevschi, S. & Danescu, R., 2017. A Multi Patch Warping Approach for Improved Stereo Block Matching. *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP)*. SciTePress, pp. 459-466. DOI: 10.5220/0006134104590466

[9] Menze, M. & Geiger, A., 2015. KITTI 2015 Dataset. *Karlsruhe Institute of Technology*. Available at: http://www.cvlibs.net/datasets/kitti/eval_scene_flow.php?benchmark=stereo.

[10] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," *Advances in Neural Information Processing Systems*, vol. 27, 2014

[11] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, pp. 2650–2658, 2015.

[12] Y. Cao, Z. Wu, and C. Shen, "Estimating depth from monocular images as classification using deep fully convolutional residual networks," IEEE Transactions on Circuits and Systems for Video Technology, vol. 28 no. 11, pp. 3174–3182, 2017.

[13] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 2002–2011

[14] S. F. Bhat, I. Alhashim, and P. Wonka, "Adabins: Depth estimation using adaptive bins," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 4009-4018

[15] Z. Li, X. Wang, X. Liu, and J. Jiang, "Binsformer: Revisiting adaptive bins for monocular depth estimation," IEEE Transactions on Image Processing, vol. 33, pp. 3964-3976, 2024

[16] S. F. Bhat, I. Alhashim, and P. Wonka, "Localbins: Improving depth estimation by learning local distributions," in European Conference on Computer Vision. Springer, 2022, pp. 480-496

[17] S. Shao, Z. Pei, X. Wu, Z. Liu, W. Chen, and Z. Li, "Iebins: Iterative elastic bins for monocular depth estimation," in Advances in Neural Information Processing Systems (NeurIPS), 2023.

[18] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias M¨uller. Zoedepth: Zero-shot transfer by combining relative and metric depth, 2023

[19] D. Zoran, P. Isola, D. Krishnan, and W. T. Freeman, "Learning ordinal relationships for mid-level vision" in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2015, pp. 388-396

[20] K. Xian, J. Zhang, O. Wang, L. Mai, Z. Lin, and Z. Cao, "Structureguided ranking loss for single image depth prediction," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 611-620

[21] T. Zhou, P. Krahenbuhl, and A. A. Efros, "Learning data-driven reflectance priors for intrinsic image decomposition," in Proceedings of the IEEE international conference on computer vision, 2015, pp. 3469-3477.

[22] W. Chen, Z. Fu, D. Yang, and J. Deng, "Single-image depth perception in the wild," Advances in neural information processing systems, vol. 29, 2016

[23] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot crossdataset transfer," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 3, 2022.

[24] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," Advances in neural information processing systems, vol. 27, 2014

[25] Z. Li and N. Snavely, "Megadepth: Learning single-view depth prediction from internet photos," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 2041-2050

[26] J. Jun, J.-H. Lee, C. Lee, and C.-S. Kim, "Depth map decomposition for monocular depth estimation," in European Conference on Computer Vision. Springer, 2022, pp. 18-34

[27] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, and H. Zhao, "Depth anything: Unleashing the power of large-scale unlabeled data," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 10 371–10 381

[28] B. Ke, A. Obukhov, S. Huang, N. Metzger, R. C. Daudt, and K. Schindler, "Repurposing diffusion-based image generators for monocular depth estimati" in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 9492-9502.

[29] M. P. Muresan, M. Raul, S. Nedevschi and R. Danescu, "Stereo and Mono Depth Estimation Fusion for an Improved and Fault Tolerant 3D Reconstruction," *2021 IEEE 17th International Conference on Intelligent Computer Communication and Processing (ICCP)*, Cluj-Napoca, Romania, 2021, pp. 233-240

[30] V. -C. Miclea, L. Miclea and S. Nedevschi, "Real-time Stereo Reconstruction Failure Detection and Correction using Deep Learning," *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Maui, HI, USA, 2018, pp. 1095-1102, doi: 10.1109/ITSC.2018.8569928

[31] R. Spangenberg, T. Langner, S. Adfeldt and R. Rojas, "Large scale Semi-Global Matching on the CPU," *2014 IEEE Intelligent Vehicles Symposium Proceedings*, Dearborn, MI, USA, 2014, pp. 195-201, doi: 10.1109/IVS.2014.6856419.

[32] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer, 2018, pp. 833–851.

[33] F. Xue, G. Zhuo, Z. Huang, W. Fu, Z. Wu and M. H. Ang, "Toward Hierarchical Self-Supervised Monocular Absolute Depth Estimation for Autonomous Driving Applications," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, NV, USA, 2020, pp. 2330-2337 .

[34] Farooq Shariq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021

[35] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. arXiv:1907.10326, 2019

[36] Sihaeng Lee, Janghyeon Lee, Byungju Kim, Eojindl Yi, and Junmo Kim. Patch-wise attention network for monocular depth estimation. In In Proceedings of the AAAI Conference on Artificial Intelligence, 2021