2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

SGM-MDE: Semi-global optimization for classification-based monocular depth estimation

Vlad-Cristian Miclea and Sergiu Nedevschi

Abstract— **Depth estimation plays a crucial role in robotic applications that require environment perception. With the introduction of convolutional neural networks, monocular depth estimation (MDE) methods have become viable alternatives to LiDAR and stereo reconstruction-based solutions. Such methods require less equipment, fewer resources and do not need additional sensor alignment requirements. However, due to the ill-posed formulation of MDE, such algorithms can only rely on learning mechanisms, which makes them less reliable and less robust.**

In this work we propose a novel method to cope with the lack of geometric constraints inherent to monocular depth computation. Towards this goal, we initially mathematically transform the feature vectors from the last layer inside a MDE CNN such that a 3D stereo-like cost volume is generated. We then adapt the semi-global stereo optimization to the aforementioned volume, global consistency of the map being ensured. Furthermore, we enhance the results by adding a sub-pixel stereo post-processing be means of interpolation functions, a larger range of depth values being obtained. Our method can be applied to any classification-based MDE, experiments showing an increase in accuracy with an additional time cost of only 8 ms on a regular GPU, making the technique usable for real-time applications.

I. INTRODUCTION

Depth estimation is one of the most important tasks in environment perception for robotic applications. A depth map containing information about the distance to each object inside the scene has to be extremely accurate, robust, dense and obtained with as few resources as possible.

The methods that are best suited for depth perception in driving scenarios use LiDAR technologies [1] [2] [3] due to their high accuracy in terms of measurements and their robustness. However, LiDAR-based methods suffer in terms of equipment cost and output density. Furthermore, their results are given in a different coordinate system than the image frame, the long acquisition period making the synchronization with other sensors problematic. Camera-based solutions based on stereo reconstruction [4] [5] [6] do not have this association problem, but generally need lots of resources (high resolution cameras, high computational cost) for good results. Generally stereo algorithms have been classified in two categories: local and global. Local methods [7] [8] [6] are less accurate, evaluating the disparity on a similarity criterion applied over small (generally maximum 5x5) support windows. On the other hand, global approaches [5] [9] evaluate the disparity of all pixels in an image as a whole by optimizing a global energy function.

The authors are with the Department of Computer Science, Technical University of Cluj-Napoca, Cluj-Napoca, Romania, E-mails: Vlad.Miclea@cs.utcluj.ro, Sergiu.Nedevschi@cs.utcluj.ro

With the apparition of deep learning, monocular depth estimation (MDE) has become a technique more and more reliable for depth perception in robotic applications. MDE algorithms [10] [11] [12] [13] rely on an image captured by a single camera, which is then used for inferring the depth map. Consequently, such methods have several advantages over their counterparts: the solution is cheaper in terms of both equipment cost and computational resources, there is no need for extra temporal alignment and it produces results with reasonable accuracy. However, since an infinity of 3D depth scenes can be produced from a single 2D image, there is a lack in geometric constraints for MDE algorithms, making such algorithms less reliable for critical application.

In this work we plan to increase the reliability of depth maps generated by MDE neural networks by augmenting the learning-based solution with information extracted via traditional stereo-like techniques. Using stereo knowledge as supervision for MDE proved to be effective [14] [15] [16] especially since real-life depth ground truths based on LiDARs (eg. Kitti [17]) are not entirely dense. However, since such approaches require stereo computation as well, they do not entirely benefit from the aforementioned single-image advantages.

Our work proposes an alternative way for MDEs to benefit from stereo information: we initially let the CNN converge towards a depth map in a supervised fashion, without interfering with the learning process. We next extract viable information from the weight vectors inside the convolutional layers, which we mathematically transform to a stereo-like 3D matching cost volume. Geometric constraints are then applied over the 3D volume be means of global optimization. In order to limit the amount of computational resources we modify the well-known semi-global optimization such that it can work in relation with the aforementioned 3D volume. Finally, we show that sub-pixel post-processing techniques can also be applied, making the final output less dependent on the depth space discretization. Thus, we find an interpolation function that refines the output, providing a more precise depth. We evaluate the proposed approach, and we highlight that the depth map is improved, a better accuracy being obtained with an additional computational cost of only 8 ms. To sum up, our original contributions are:

- A mathematical transformation of the probability volume generated by the last layer in classification-based MDE solutions such that it can be used for stereo-like optimizations;
- A novel method for introducing stereo semi-global optimization into monocular depth estimation procedures;

- An adaptation of sub-pixel post-processing to MDE solutions; This increases the range of the resulting depth map, making it independent on the space discretization technique.

The article is structured as follows: We start with presenting the state of the art in stereo and monocular depth perception. In section 3 we initially introduce a mathematical transformation of the 3D probability volume generated by the MDE, and show how the semi-global optimization is the applied. Section 4 discusses the results obtained by our new procedure with respect to other depth perception algorithms. Finally, we conclude the paper in section 5.

.

II. Related Work

A. Stereo Reconstruction

Most of the recent stereo algorithms use deep learning methods under the form of end-to-end procedures. Most of top methods on Kitti benchmark use CNNs that formulate the disparity selection using an differentiable $argmin$, modeling the problem as a regression. The method that initially proposed to use a soft fully differentiable $argmin$ function was GC-Net [18]. The method uses 3D convolutions to incorporate both geometry and context and then regresses towards the sub-pixel value of the disparity GT. Most of the other methods that currently rank high in Kitti stereo dataset fall into this category (eg. GA-Net [19], AMNet [20]). However, none of them is capable of producing disparities in real-time (most of them take around 1 second).

For real-time applications, the traditional pipeline (originally proposed by D. Scharstein and R. Szeliski [4]) provides several solutions presenting a viable accuracy-sepeed tradeoff. By following the traditional approach, stereo reconstruction is divided into four main phases: cost computation, aggregation, optimization, refinement. Each phase is responsible for solving a particular sub-problem. The global optimization is the key part from the four phases, an energy function incorporating both a cost term and a smoothness term (which adds geometrical constraints to the problem) being defined on all the pixels of the image. Thus, local ambiguities caused by photometric variations, reflections or occlusions are alleviated.

Semi-global matching (SGM) [5] is one of the most robust optimization algorithms, ensuring close-to global consistency while consuming a reasonable amount of resources. The algorithm behaves like a global algorithm, performing an energy minimization on several (generally 8) 1D paths crossing each pixel and thus approximating the 2D image. There are various implementations of SGM on different platforms CPU [21], GPU [22] or FPGA [23], most of them obtaining real-time performances. Other similar optimization methods like Graph Cuts [24], Belief propagation [25], or Total Variation [9] are very expensive in terms of resources.

B. Supervised Monocular Depth Estimation

Since an infinity of 2D depth maps can be generated from a 3D scene, monocular depth estimation approaches try first to understand locally the relations between objects and then exploit these cues [10]. Learning the scene reprezentation for this requires convolutional neural networks. Most of the approaches in the MDE category use encoder-decoder arcitectures, minimizing a mean squared error-type of loss [12] [26]. The most representative ways to capture object relation-representation is by either extracting features at multiple scales (under the form of a pyarmid)[27], by using dilated convolutions [28] or by using attentional gating [11] (features from initial layers (coarse) being later plugged in for a better filtering of relevant information).

Although most of the MDE algorithms directly regress the (fractional) depth map, Xian et al. [29] proved that better results are obtained when the MDE problem is seen as a classification rather than a regression (Table 3 and 4 in [29]). In the classification case the network converges faster and it produces more robust results. However, classification assumes a fixed number of depth classes, meaning that the depth interval has to be discretized. A common way to discretize the spatial depth interval is to use uniform discretization [30] [31], which equally accounts for the near-side classes and the far-side ones.

DORN [13] proposes an improved discretization technique, using the log-space (more classes in the near-space). Such an approach is capable to more accurately predict short-range and near-range depth values, which constitute the vast majority of points inside the dataset (because of this DORN ranked 1st for more than 1 year on Kitti). Another important factor for the success of DORN is that its loss also accounts for neighboring classes. Such a loss is based on ordinal regression (or ordinal classification) and it provides a higher accuracy and faster convergence. There are other similar approaches (such as SORD [32]) that also perform very well by understanding of the intraclass and interclass ordinal relationships.

C. Stereo-based monocular depth estimation

Due to the lack of sufficient training data, supervised MDE methods generally perform worse than stereo counterparts. In order to alleviate this, semi-supervised [14] [15] and unsupervised [16] learning-based MDE approaches have emerged. These methods combine the depth prediction of two supervised networks (for left and right) with their corresponding stereo-based disparity information in a single loss, capable to produce a better depth map. Other works such as [33] [34] use self-supervising techniques by processing a single image to obtain a second synthetic view (similar to the right view in stereo). A CNN is then trained on the two images for computing a disparity map.

Other MDE methods try to obtain higher accuracy not by depth supervision, but by augmenting convolutional networks with 3D constraints. Such methods either use conditional random fields or surface normals [35] to capture and exploit the geometry of the scene. The formulation of these approaches is cumbersome (the geometrical constraints are included in the loss function) and they also need a high amount of computational resources, which makes them unfeasible

for real-life applications. Instead, we prefer to augment MDE methods with geometric constraints in a lightweight, algorithmically-friendly way, which can also lead to real-time implementations.

III. Semi-Global Optimization for MDE

We will initially show how to generate a stereo-like cost volume from a MDE network. We will highlight the mathematical transformations required in order to transform the probabilities given by the networks either in classification-like CNNs and in ordinal regression-type CNNs. Next we will present how we can adapt the semi-global optimization to these 3D volumes. We will then show how sub-pixel accurate values can be obtained, by a post-processing using interpolation functions. The overall architecture of the system is presented in Fig. 1.

A. From MDE probability volume to 3D stereo cost volume

In this part we will show how a 3D stereo-like cost volume can be generated. The only constraint for this generation is that the MDE problem is modeled as either as a classification, or an ordinal regression, but not as a regular regression. This is because simple regression networks generate a 2D map in the last layer, not a 3D one. As previously mentioned, the classification-type provides better results, but is more costly.

a) Case 1: Simple Classification MDE: For a simple classification-based MDE network, the discretization of the depth interval will guide the optimization process. A limited number of depth values are inferred and each such depth value might be considered as an optimal position. Thus, we will focus on the last layer of the network, which generates a $width \times height \times depth\_classes$ probability volume, where $width$ and $hiehgt$ are the image dimensions. The number of classes $depth\_classes$ will be directly dependent on the depth interval discretization, and it can be chosen by considering:

$$depth\_classes = \frac{max\_depth}{samp\_f} \tag{1}$$

where max_depth is the maximum depth allowed for measurement (generally 80, for driving applications) and $samp\_f$ is the sampling factor for the discretization (how large is a depth interval). For instance, if $samp\_f = 1$, then all the integer metric distances are considered as possibilities.

For a faster convergence, the last layer in most of the classification-based MDE approaches is generally a $softmax$, assigning a probability for each position to belong to a specific depth class. During the evaluation, the best depth value is selected by finding the position of the maximum probability.

$$dep_{clasif}(p) = argmax_d(C_{softmax}(p, d)) \tag{2}$$

where $C_{softmax}$ is the 3D probability volume and $argmax$ chooses the position of the largest probability for the pixel positioned at (p).

Now, if we wan to change the perspective towards a traditional stereo, we can then observe that the chosen depth/disparity will be given by the argmax function (position of the maximum probability); Traditional stereo approaches use argmin function, for finding the minimum cost. Thus, in order to apply the same depth/disparity selection procedure, we will need to compute "inverse probability" (cost) volume. This will will be generated by:

$$C_{i\_prob}(p, d) = \alpha \times (1 - C_{softmax}(p, d)) \tag{3}$$

where α is a parameter that normalizes the cost values such that integer values are generated. In this case, $\alpha = 255$. Now a simple $argmin$ selection will choose the optimal depth:

$$dep_{i\_prob}(p) = argmin_d(C_{i\_prob}(p, d)) \tag{4}$$

b) Case 2: Ordinal Regression MDE: State of the art benchmarks show that ordinal regression (also known as ordinal classification) outperforms simple classification counterparts, approaches such as DORN [13] or SORD [32] being very accurate. Ordinal regression can be seen as a more complex classification, in which values from neighboring depths are accounted in the final loss. From our stereo-cost generation perspective, the mathematical transformations required for building the cost volume will be more complicated.

We will start from the $softmax$ function (like in Case I) from [13] (eq. (3) in their paper). During the evaluation, the best depth value is now not selected by finding the position of the maximum probability, but rather by finding the median value of the distribution. According to eq (5) from [13], the chosen depth is given by the position of the last value whose probability is not smaller than 0.5:

$$dep_{OR}(p) = arg_d(C_{sf\_or}(p, d) \geq 0.5) \tag{5}$$

This softmax probability function ($C_{sf\_or}(p, d)$) (eq. 3 from [13]) looks similar to a cumulative distribution function: the distribution for a given spatial location varies according to the inverse logistic function (Fig. 2 up), where the vertical line highlights the position of the chosen depth. This logistic function is continuous and differentiable on interval $[0..depth\_classes]$. Our goal is to transform this distribution such that the same position is given, but its form can be exploited by stereo-like optimizations (similar to Fig. 2 down). Thus, in order to obtain a stereo cost-like distribution, we will have to mathematically transform $C_{sf\_or(p,d)}$ to a new curve distribution $C_{i\_prob}(p, d)$, which has the following properties:

- $C_{i\_prob}(p, d)$ is differentiable on the interval $[0, depth\_classes]$
- The minimum position for $C_{i\_prob}(p, d)$ is dep, i.e. $C_{i\_prob}(p, dep)) \leq C_{i\_prob}(p, d)$, $\forall d$. Ideally, $C_{i\_prob}(p, dep) \approx 0$.

.

By considering these two properties, we propose the following inverse probability (cost) function for the ordinal regression case:

$$C_{i\_prob}(p, d) = [C_{sf\_or}(p, d) - C_{sf\_or}(p, dep)]^2 \tag{6}$$

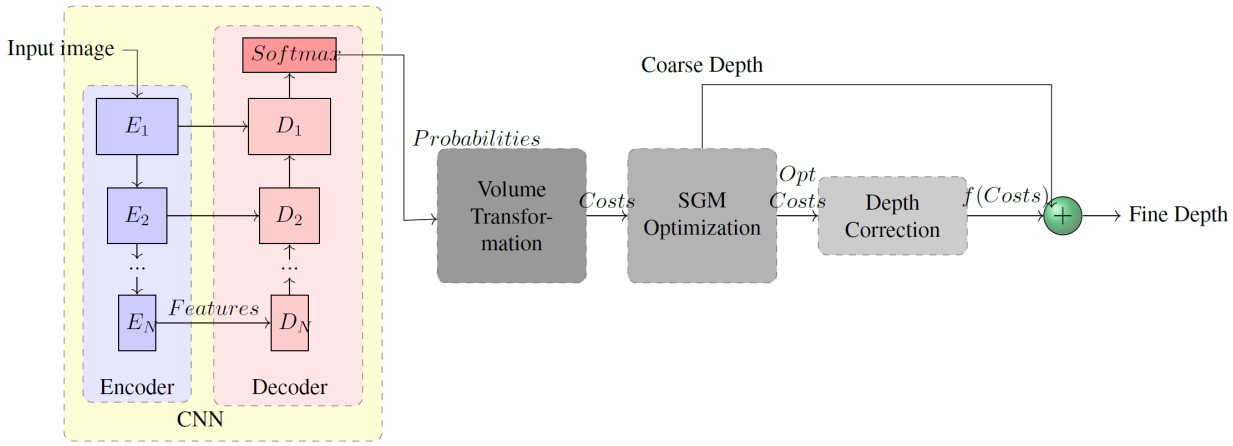


Fig. 1: Architecture of the MDE System

We can easily prove that our newly generated cost curve $C_{i\_cost}(p,d)$ follows the aforementioned properties. Firstly, if $d \to dep \implies C_{i\_prob}(p,d) \to 0$. We are also interested in the form of the new function. This can be seen if we expand and then derive $C_{i\_cost}(p,d)$ with respect to d:

$$\begin{aligned} C_{i\_prob}(p,d) &= C^2_{sf\_or}(p,d) - 2C_{sf\_or}(p,d)C_{sf\_or}(p,dep) \\ &\quad + C^2_{sf\_or}(p,dep) \end{aligned} \tag{7}$$

$$\begin{aligned} C'_{i\_prob}(p,d) &= 2C_{sf\_or}(p,d)C'_{sf\_or}(p,d) \\ &\quad - 2C'_{sf\_or}(p,d)C_{sf\_or}(p,dep) \\ &= 2C'_{sf\_or}(p,d)[C_{sf\_or}(p,d) - C_{sf\_or}(p,dep)] \end{aligned} \tag{8}$$

Approximating the derivative around dep gives that:

$$C'_{i\_prob}(p,dep) \approx 2C'^2_{sf\_or}(p,dep) \tag{9}$$

As mentioned, the minimum point for $C_{i\_prob}(p,d)$ will be dep. A second observation that we can draw from the previous expression is that our newly generated distribution will increase/decrease quadratically with respect to the original distribution (a larger inter-class difference is introduced).

B. Semi-global Optimization for MDE

Once the stereo-like cost volume is generated, we can use the SGM optimization technique. The most critical part in SGM is the penalty selection: good values for P_1 – penalty for small disparity changes and P_2 – penalty for large disparity disruptions are necessary. While some methods train particular penalties for each pixel (generated through deep learning [36]), other adapt them to surface types [37]. In our case, exhaustive tests show that penalty selection procedure does not depend on the surface type but rather on the way in which the softmax probability is distributed across the depth space. There are situations in which two neighboring pixels present two distributions parametrically different. Therefore, a normalization factor has to be introduced into the SGM formulation. This will be done only for the ordinal regression case, since the classification case is straightforward (a regular SGM formula can be used).

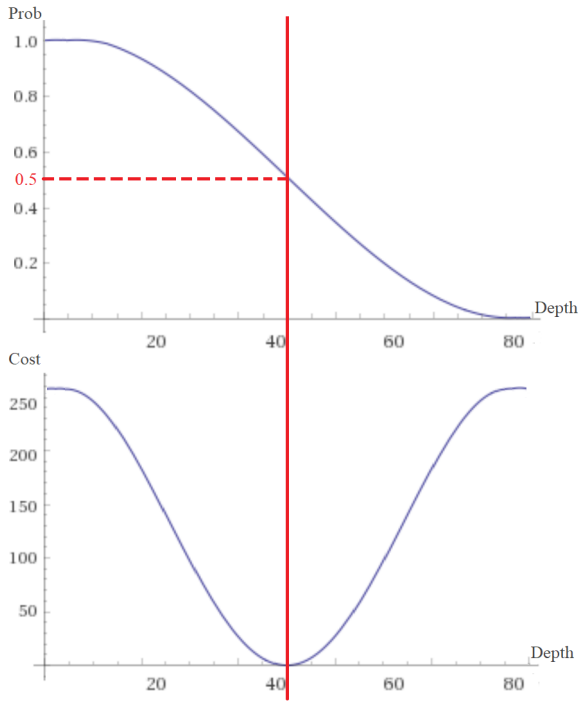


Fig. 2: Probability distributions for ordinal regression softmax $C_{sf\_or}(p)$ (up) and obtained cost distribution $C_{i\_prob}(p)$ (down); The the position of the minimal cost is highlighted

For the ordinal regression we observe two corner cases, which determine the way in which the penalties have to be adapted:

- Probabilities from $C_{sf\_or}(p,d)$ decrease linearly (similarly to the ones in Fig. 2 up). As a consequence, the distribution for $C_{i\_prob}(p,d)$ is more flatten. In this case neighboring probability values are close to the selected one, so there should be a small value for penalty P1;
- Probabilities from $C_{sf\_or}(p,d)$ decrease abruptly (in a step-like distribution). Consequently, $C_{i\_prob}(p,d)$ has a peak that overlooks the neighbors. In this case the

penalties have to be larger, to compensate for the class variation.

The formulation of the inverse probability cost can directly provide an adaptation of the SGM formula such that both of these marginal cases are covered. The derivative of the function directly determines the amount by which the penalty is emphasized. So, we can directly include this slope factor into SGM:

$$\begin{aligned} C_{sgm}(p,d) = C_{i\_prob}(p,d) + min(&C_{sgm}(p-r,d_p), \\ &C_{sgm}(p-r,d_p-1) + \beta C'_{i\_prob}(p,d) \times P_1, \\ &C_{sgm}(p-r,d_p+1) + \beta C'_{i\_prob}(p,d) \times P_1, \\ &min_{k \in D} C_{sgm}(p-r,k) + \beta C'_{i\_prob}(p,d) \times P_2) \end{aligned} \quad (10)$$

where D = $[0..depth\_classes]$, β is a scaling factor, r is the direction for optimization, and $C'_{i\_prob}(p,d)$ gives the direction and the amount by which penalty is modified. Finally, the optimized cost is computed by:

$$C_{final}(p,d) = \sum_r C_{sgm}(p,d_p) \quad (11)$$

The winner takes all (WTA) selection technique is then applied, the depth being generated as:

$$dep_{after\_sgm}(p) = argmin_d(C_{sgm}(p,d)) \quad (12)$$

C. Sub-pixel refinement

As previously mentioned, selection will extract integer values, on the positions of minimal costs. Since the range of these values is limited by the maximum number of classes, sub-pixel correction creates the opportunity to slightly modify the final value in the correct direction. This is also done in traditional stereo approaches, which formulate the final depth as:

$$d_{SubPx}(p) = d_{Int}(p) + f(c_{d-1}, c_d, c_{d+1}) \quad (13)$$

where D_{Int} is the integer disparity and c values are matching costs (taken from the cost volume generated in cost computation and optimization steps) neighboring the chosen integer disparity and f is a function which should redirect the disparity towards the correct fractional depth. These parameters provide enough information for an accurate estimation with sub-pixel precision.

The main problem becomes to infer the proper function, based on values extracted from the cost volume. A function that wrongly refines the initial map might lead to the so called "pixel-locking effect": an overcrowding of disparities towards integer values. Methods such as [38] [39] [40] use function fitting mechanisms to learn the correct interpolation function. We can also adapt the stereo mechanism to MDE by expressing the final depth value as:

$$dep_{final}(i,j) = dep_{after\_sgm}(p) + samp\_f \times f(c_{d-1}, c_d, c_{d+1}) \quad (14)$$

where c_d is the SGM cost at the chosen depth, while c_{d-1} and c_{d+1} represent the neighboring costs and f is a function which should redirect the depth towards the correct depth class. This formulation is similar to stereo sub-pixel, with the exception that MDE also includes the sampling factor. By applying the same function fitting procedure as in [41], we can obtain new functions: (eq. 15) for the classification case (for uniform distribution) and (eq. 16) for ordinal regression case (non-uniform distribution).

$$f(c_{d-1}, c_d, c_{d+1}) = sin(arctan(\frac{c_{d+1} - c_d}{c_d - c_{d-1}} \times \pi/2) \times \pi/2) \quad (15)$$

$$f(c_{d-1}, c_d, c_{d+1}) = 1 - cos(\frac{c_{d+1} - c_d}{c_d - c_{d-1}} \times \pi/2) \quad (16)$$

IV. Evaluation

A. Datasets and Training

a) Datasets: Since we use supervised learning techniques for optimization, a reliable dataset is required. Because the scope of our work is automated driving perception, Kitti [42] is the best option for us. Thus, we use the Kitti raw dataset images as inputs and depth prediction ground truths for the validation of the CNNs.

b) Methods for comparison and Implementation details: We evaluate:

- A classification-based network – For this case we have also implemented our own classification-based MDE estimation; The method uses Darknet-53 feature extractor (from Yolo V3 [43]) and it uses cross-entropy as loss for learning. 80 classes have been used for the last layer; We chose this lightweight feature extractor due to its real-time capabilities; For this case we use multiple CNN implementations: we variate the sampling factor, to obtain various depth intervals. We have trained this method using the Darknet framework, using the C++ and CUDA languages.
- An ordinal regression-based network – For this case we use DORN: the state of the art on Kitti, that uses an ordinal regression loss; For this we use the code provided by the authors. However, there is no code for training, so we use the provided trained model. The code is written in PyTorch. Predicting the depth using DORN takes around 0.5 s.

We have implemented SGM variants for both these approaches. We used CUDA and C++ languages for the first case, while for the second we used a Python implementation. Both these implementations run directly on a regular GPU (NVidia 1080 Ti), our goal being also to obtain a good time performance. For post-processing, the algorithms use the interpolation functions presented in (eq. 15-16).

B. Results

1) Accuracy Improvements:

a) SGM Optimization: We evaluate here the improvements provided by the SGM optimization. Initially we show the results our optimization method obtains for the entire driving scene with respect to the baselines and to other MDE methods. Thus, we evaluate the YoloV3-based depth classification (close to real-time), with and without the optimization,

for the case in which the sampling factor $samp\_f = 2$ and DORN [13]. We also include a regular regression implementation for the Yolo-based network and an implementation for the CNN proposed by Eigen et. al. [12]. Results obtained by other methods can easily extracted from Kitti benchmark and compared with this work. We evaluate the depth estimation metrics from Kitti: the relative absolute error (absErrorRel), the squared relative error (sqErrorRel), the inverse root mean squared error (iRMSE) and the Scale Invariant Log (the primary metric in Kitti for depth prediction)

In terms of precision, we observe that the overall accuracy (for the entire image) is improved. The margin of improvements with respect to DORN is smaller, since this network is already very accurate, but still sufficiently large to justify adding the optimization. The improvements over the classical classification approach are more significant. Numerical results are presented in Table I. The validity of our method can also be visually observed in Figure 4, which depicts the results obtained by our refinement method when applied over DORN (because of space limits we restrict to visually depict only this case). Although both depth maps (c) and d)) seem quite similar, images containing the depth error (e) and f)) clearly show the differences between the two. In these images, pixels that are closer to GT are depicted in blue, while erroneous pixels are depicted in red/orange. It can also be visually seen that many more points are more accurately computed, the depth map in the entire scene being more reliable.

b) Long-range accuracy: We also examine the results obtained by our method when focusing on the long-range part of the scene because the sub-pixel post-processing part is particularly interested in dealing with large distances. Thus, we consider the points whose value is between 50 and 80. We evaluate this according to the general depth estimation metrics on Kitti.

We test two variants for both types of CNNs: one without any correction, and one with an optimal correction (denoted +PostProc). The Yolo-based classification problem uses a sampling factor of 1 (80 classes) uniformly distributed, while DORN uses 142 classes, unevenly distributed. Results show that the refinement technique is beneficial for objects at large distances for both MDE algorithms. The largest improvement is obtained in the case of DORN. This is somehow expected, since DORN overlooks objects in the far-side (due to the log discretization). Numerical results can be seen in Table II.

2) Speed:

a) Sampling factor variation: We are interested in seeing how our method behaves when we variate the number of classes: this gives the way in which we discretize the depth space. We test here multiple CNN models, which correspond to the following number of depth classes/intervals: 80, 40, 20, 10 and 5. We have chosen these values because Kitti max depth value is around 80 (depths are in range 0-80). Thus, we use five possible sampling factors: 1, 2, 4, 8 and 16. We test three methods: without any optimization, with SGM optimization and corrected using the proposed interpolation function. The first important aspect we observe is that runtime decreases almost exponantially with depth class reduction, until the encoder part becomes the critical part in time consumption. Error, on the other hand has a much smaller increment. The results can be see in Fig. 3. It can bee seen that if we use large sampling factors, the method runs in real-time. However, decent performance is obtained only for sampling factors smaller than 8 (more than 10 depth classes are required). Another important observation is that post-processing becomes more relevant when the sampling factor is larger (its burden related to depth distribution becomes more relevant). Optimization, on the other hand, produces similar error reduction for any sampling factor choice.

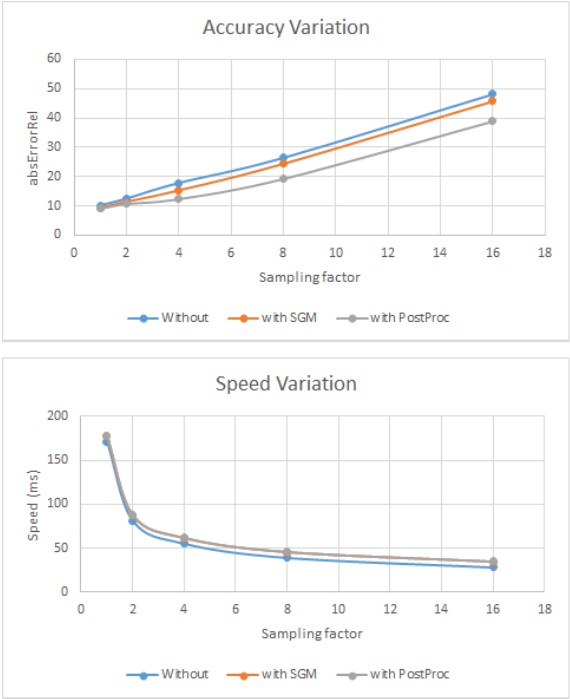


Fig. 3: Accuracy vs Speed variation for different sampling factors

b) Overall time performance: In this part we are interested in evaluating the additional time required for each step in our system: for the initial CNN, the mathematical transformation of the cost, SGM optimization and the interpolation function. Numerical results can be seen in Table III. It can be seen that the C++/CUDA implementation for the network with a sampling factor of 8 is really fast, resulting in an overall time of 46 ms. This proves that our method does not need too many resources, the number of computations being relatively small. Overall, for the C++/CUDA case, only an additional time of 8 ms is required (1 for cost transformation and for post-processing and 6 ms for SGM optimization). For the Python case our algorithm is slower, taking an additional 75 ms. All in all, we can highlight that our method can run in (close-to) real-time, while obtaining a decent accuracy.

TABLE I: Performance of the MDE correction methods for all pixels in Kitti images

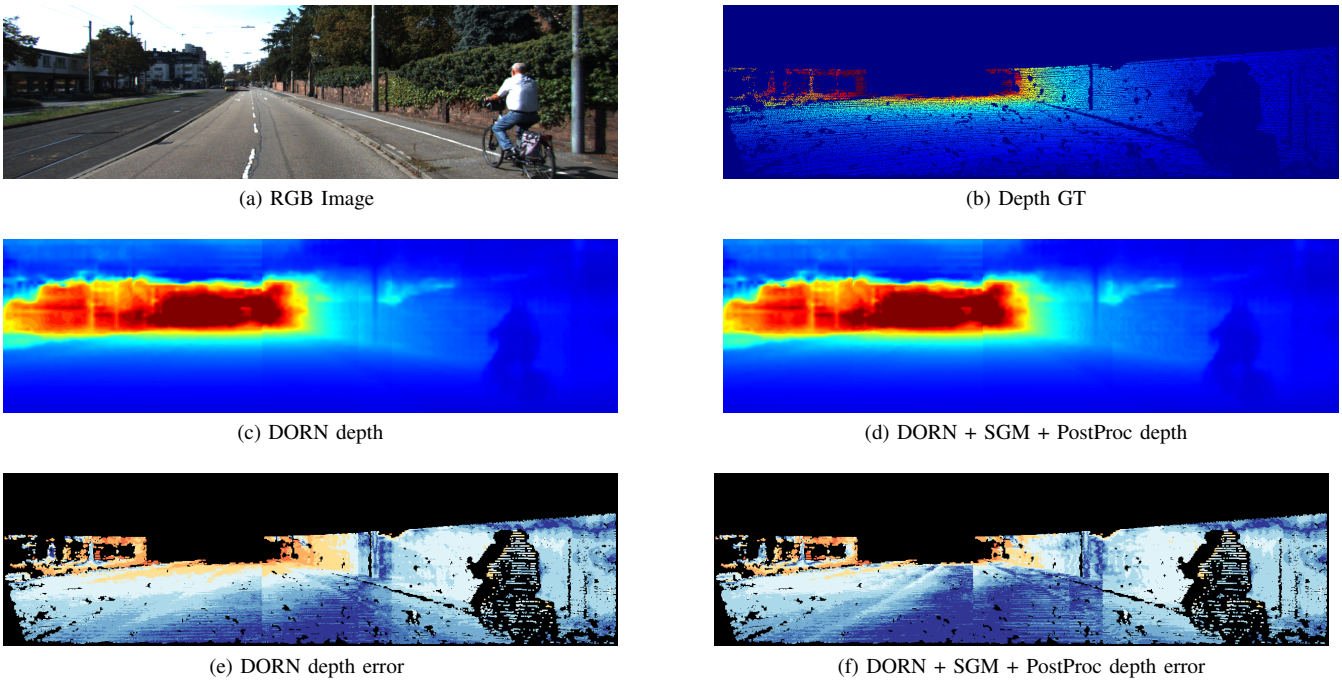
Method	Platform	SILog	sqErrorRel	absErrorRel	iRMSE
Yolo-based classif;	GPU (C++/CUDA)	16.37	5.12 %	12.24 %	17.28
Yolo-based classif; (+SGM)	GPU (C++/CUDA)	13.98	4.11 %	10.14 %	14.93
DORN [13]	GPU (Python)	12.01	3.53 %	7.55 %	11.98
DORN (+SGM) [13]	GPU (Python)	10.68	2.93 %	6.39 %	10.06
Yolo-based regression	GPU (C++/CUDA)	18.07	5.68 %	18.33 %	20.98
Eigen [12]	GPU (Python)	20.22	6.02 %	19.17 %	22.92

TABLE II: Results obtained with our depth correction on long-range MDE Kitti images ($Depth \in [50, 80]$)

Method	SILog	sqErrorRel	absErrorRel	iRMSE
Yolo-based classif	17.47	4.43 %	18.58 %	21.98
Yolo-based classif (+PostProc)	17.01	4.20 %	17.89 %	20.15
DORN [13]	17.55	4.53 %	18.78 %	22.89
DORN (+PostProc) [13]	15.22	3.68 %	16.33 %	20.14

TABLE III: Time performance for each step of the method for Kitti images

Method	CNN	+ Volume Transf.	+ SGM	+ PostProc
Yolo-based $samp\_f = 8$	38 ms	39 ms	45 ms	46 ms
DORN [13]	501 ms	504 ms	572 ms	576 ms



(a) RGB Image (b) Depth GT

(c) DORN depth (d) DORN + SGM + PostProc depth

(e) DORN depth error (f) DORN + SGM + PostProc depth error

Fig. 4: Results obtained with our depth correction approach for Kitti images

V. CONCLUSIONS

We have presented here a method to improve the accuracy of supervised monocular depth estimation algorithms by inserting stereo-like geometric constraints into the MDE algorithms. In order to do this, we initially show a method to mathematically transform the 3D probability volume generated by classification and ordinal regression-based approaches to a 3D stereo-like cost volume. We then insert a global optimization based on semi-global matching over the aforementioned volume. Finally, we find an interpolation function that refines the output, providing a more precise depth by increasing the range. The method proves that a better accuracy is obtained, by also preserving the real-time capabilities for the underlying algorithms (only an additional time of 8 ms is required for the optimization).

We plan to continue our work with respect to depth perception by adapting other stereo reconstruction techniques to monocular counterparts. For instance, we plan to research new ways to help MDE methods based on regular regression benefit from scene geometry.

ACKNOWLEDGEMENT

This work was supported by UEFISCDI (Romanian National Research Agency) in the national research project SEPCA (Semantic Visual Perception and Integrated Control

for Autonomous Systems), project code PN-III-P4-ID-PCCF-2016-0180.

REFERENCES

[1] J. Hecht, "Lidar for self-driving cars," *Opt. Photon. News*, vol. 29, no. 1, pp. 26–33, Jan 2018. [Online]. Available: http://www.osa-opn.org/abstract.cfm?URI=opn-29-1-26

[2] K. Lim, P. Treitz, M. Wulder, B. St-Onge, and M. Flood, "Lidar remote sensing of forest structure," *Progress in Physical Geography*, vol. 27, pp. 88–106, 03 2003.

[3] W. Maddern and P. Newman, "Real-time probabilistic fusion of sparse 3d lidar and dense stereo," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 2181–2188.

[4] 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

[5] 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.

[6] 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.

[7] 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.

[8] M. Humenberger, C. Zinner, M. Weber, W. Kubinger, and M. Vincze, "A fast stereo matching algorithm suitable for embedded real-time systems," *Comput. Vis. Image Underst.*, vol. 114, no. 11, pp. 1180–1202, Nov. 2010. [Online]. Available: http://dx.doi.org/10.1016/j.cviu.2010.03.012

[9] R. Ranftl, S. Gehrig, T. Pock, and H. Bischof, "Pushing the limits of stereo using variational stereo estimation," in *Intelligent Vehicles Symposium (IV), 2012 IEEE*, June 2012, pp. 401–407.

[10] A. Bhoi, "Monocular depth estimation: A survey," *CoRR*, vol. abs/1901.09402, 2019. [Online]. Available: http://arxiv.org/abs/1901.09402

[11] S. Kong and C. C. Fowlkes, "Pixel-wise attentional gating for parsimonious pixel labeling," *CoRR*, vol. abs/1805.01556, 2018. [Online]. Available: http://arxiv.org/abs/1805.01556

[12] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," *CoRR*, vol. abs/1406.2283, 2014. [Online]. Available: http://arxiv.org/abs/1406.2283

[13] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[14] Y. Kuznietsov, J. Stckler, and B. Leibe, "Semi-supervised deep learning for monocular depth map prediction," 07 2017, pp. 2215–2223.

[15] A. J. Amiri, S. Y. Loo, and H. Zhang, "Semi-supervised monocular depth estimation with left-right consistency using deep neural network," *CoRR*, vol. abs/1905.07542, 2019. [Online]. Available: http://arxiv.org/abs/1905.07542

[16] C. Godard, O. M. Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6602–6611, 2016.

[17] 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.

[18] 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

[19] F. Zhang, V. Prisacariu, R. Yang, and P. H. Torr, "Ga-net: Guided aggregation net for end-to-end stereo matching," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[20] X. Du, M. El-Khamy, and J. Lee, "Amnet: Deep atrous multiscale stereo disparity estimation networks," *CoRR*, vol. abs/1904.09099, 2019. [Online]. Available: http://arxiv.org/abs/1904.09099

[21] 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.

[22] 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.

[23] 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.

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

[25] 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

[26] J. H. Lee, M. Han, D. W. Ko, and I. H. Suh, "From big to small: Multi-scale local planar guidance for monocular depth estimation," *CoRR*, vol. abs/1907.10326, 2019. [Online]. Available: http://arxiv.org/abs/1907.10326

[27] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," *CoRR*, vol. abs/1612.03144, 2016. [Online]. Available: http://arxiv.org/abs/1612.03144

[28] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *CoRR*, vol. abs/1511.07122, 2015. [Online]. Available: http://arxiv.org/abs/1511.07122

[29] R. Li, K. Xian, C. Shen, Z. Cao, H. Lu, and L. Hang, "Deep attention-based classification network for robust depth prediction," *CoRR*, vol. abs/1807.03959, 2018. [Online]. Available: http://arxiv.org/abs/1807.03959

[30] W. Yin, Y. Liu, C. Shen, and Y. Yan, "Enforcing geometric constraints of virtual normal for depth prediction," in *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[31] 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, 2018.

[32] R. Daz and A. Marathe, "Soft labels for ordinal regression," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019, pp. 4733–4742.

[33] F. Tosi, F. Aleotti, M. Poggi, and S. Mattoccia, "Learning monocular depth estimation infusing traditional stereo knowledge," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[34] Y. Luo, J. Ren, M. Lin, J. Pang, W. Sun, H. Li, and L. Lin, "Single view stereo matching," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 155–163.

[35] W. Yin, Y. Liu, C. Shen, and Y. Yan, "Enforcing geometric constraints of virtual normal for depth prediction," in *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[36] 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.

[37] V. Miclea and S. Nedevschi, "Real-time semantic segmentation-based stereo reconstruction," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2019.

[38] I. Haller and S. Nedevschi, "Design of interpolation functions for subpixel-accuracy stereo-vision systems," *Image Processing, IEEE Transactions on*, vol. 21, no. 2, pp. 889–898, Feb 2012.

[39] M. Shimizu and M. Okutomi, "Sub-pixel estimation error cancellation on area-based matching," *International Journal of Computer Vision*, vol. 63, no. 3, pp. 207–224, 2005.

[40] 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.

[41] V.-C. Miclea, C.-C. Vancea, and S. Nedevschi, "New sub-pixel interpolation functions for accurate real-time stereo-matching algorithms," in *Intelligent Computer Communication and Processing (ICCP), 2015 IEEE International Conference on*, Sept 2015, pp. 173–178.

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

[43] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *CoRR*, vol. abs/1804.02767, 2018. [Online]. Available: http://arxiv.org/abs/1804.02767