LiDAR - Stereo Camera Fusion for Accurate Depth Estimation

Hafeez Husain Cholakkal
Dept. of Mechanical Engineering
Politecnico di Milano
Milano, Italy
hafeezhusain.cholakkal@polimi.it

Simone Mentasti
DEIB
Politecnico di Milano
Milano, Italy
simone.mentasti@polimi.it

Mattia Bersani
Dept. of Mechanical Engineering
Politecnico di Milano
Milano, Italy
mattia.bersani@polimi.it

Stefano Arrigoni
Dept. of Mechanical Engineering
Politecnico di Milano
Milano, Italy
stefano.arrigoni@polimi.it

Matteo Matteucci
DEIB
Politecnico di Milano
Milano, Italy
matteo.matteucci@polimi.it

Federico Cheli
Dept. of Mechanical Engineering
Politecnico di Milano
Milano, Italy
federico.cheli@polimi.it

***Abstract*—Dense 3D reconstruction of the surrounding environment is one the fundamental way of perception for Advanced Driver-Assistance Systems (ADAS). In this field, accurate 3D modeling finds applications in many areas like obstacle detection, object tracking, and remote driving. This task can be performed with different sensors like cameras, LiDARs, and radars. Each one presents some advantages and disadvantages based on the precision of the depth, the sensor cost, and the accuracy in adverse weather conditions. For this reason, many researchers have explored the fusion of multiple sources to overcome each sensor limit and provide an accurate representation of the vehicle's surroundings. This paper proposes a novel post-processing method for accurate depth estimation, based on a patch-wise depth correction approach, to fuse data from LiDAR and stereo camera. This solution allows for accurate edges and object boundaries preservation in multiple challenging scenarios.**

***Index Terms*—Sensor fusion, depth, stereo camera, LiDAR**

I. Introduction

Over the last couple of decades, the world has witnessed an increased influence of automation in daily life. This ranges from a simple thermostat controlling the temperature to most recently automated vehicles and humanoid robots. In the field of the automobile industry, the Advanced Driver-Assistance System (ADAS) evolved from being merely an indicator of potential malfunction to completely taking control of the driving task. Effective perception of the surrounding environment is essential in enabling the system to perform any of these functions. Although a vast majority of earlier driver-assistance systems solely demanded a visual feed, like in the case of reverse parking camera, there is a growing need for depth information of the scene for various applications. High-resolution depth maps are fundamental for modern ADAS applications like obstacles [1], pedestrian [2], and vehicle detection [3]. Multi-object tracking is another task where dense depth maps are utilized [4]. Recently Prakash et al. [5] demonstrated the use of depth images in tele-operated driving to mitigate the effect of time delays.

Depth maps and 3D reconstruction of a vehicle's environment can be retrieved with different sensors, like cameras, LiDARs, and radars. Each of these sensors presents some disadvantages; depth from stereo vision is exceptionally dense, but also noisy, and the maximum range is generally lower than 25 meters. LiDARs, on the other hand, is accurate, but the point cloud they provide is significantly sparser compared to a camera. Finally, radar offers some unique information, like obstacles speed, but generally has a limited field of view on the vertical axis and are also subject to noise and false positives due to reflection. For this reason, the most adopted solution is to merge data coming from multiple sources to overcome the single sensor limitations. Depending on the application scenario, different approaches to the problem can be used. What we are proposing in this paper is an innovative pipeline for LiDAR-stereo camera fusion, based on a patch-wise depth correction approach. In our solution, the sparse but accurate range data available from the LiDAR act as references for improving the stereo camera's depth estimation. The continuity of the depth map is restored with a Bilateral Filter while preserving edges and object boundaries. Moreover, to maintain computational efficiency, the algorithm is implemented in a parallel computing architecture of modern Graphics Processing Units (GPU), making it suitable for real-time applications. To conclude, our approach has been tested in a real scenario using an experimental vehicle, equipped with a Velodyne VLP-16 LiDAR and a ZED stereo camera from Stereolabs. Then, to provide a quantitative analysis, it has been validated using a dataset from the KITTI Vision Benchmark Suite [6].

This paper is structured as follows: in Section II, we illustrate the current state of the art regarding depth estimation

Supported by project TEINVEIN: TEcnologie INnovative per i VEicoli Intelligenti, CUP (Codice Unico Progetto - Unique Project Code): E96D17000110009 - Call "Accordi per la Ricerca e l'Innovazione", cofunded by POR FESR 2014-2020 (Programma Operativo Regionale, Fondo Europeo di Sviluppo Regionale – Regional Operational Programme, European Regional Development Fund).

978-8-8872-3749-8

and depth sensor fusion. In Section III, we report the approach used to perform our system calibration, a fundamental task to guarantee an accurate sensor fusion. Then in Section IV, we describe the proposed pipeline for depth fusion, and in Section V, we illustrate the algorithm's performance on a publicly available dataset.

II. Related works

Dense 3D reconstruction approaches from multiple sensor sources can be arranged in three categories: Image-guided depth upsampling, Assisted stereo matching, and Machine learning approaches.

The first one addresses the problem of upsampling sparse depth data, with the help of images of the scene having the target resolution. Yang et al. [7] proposed an iterative approach to refine low-resolution range image, in terms of both its spatial resolution and depth precision, using high-resolution color images as reference. Of particular interest is the usage of a bilateral filter based on the assumption of depth similarity for similar colored pixels in a region. Chan et al. [8] use a similar approach to upsample the depth map aided by a video stream, with particular emphasis on preventing the texture of a color image from being copied to smooth areas of a noisy depth map. In their work, they extend the Joint Bilateral Upsampling approach (Kopf et al. [9]) and introduce a new noise-aware filter. The proposed solution adaptively dampens the influence of color similarity in-depth regions, which are likely to be geometrically smooth but heavily noise affected. Diebel and Thrun [10] proposes a Markov Random Field (MRF) method integrating data from low-res depth image and high-res camera image, in a similar way to the solution also suggested by Gould et al. [11]. Reulke [12] investigates methods of combining range information from a Photonic Mixer Device (PMD) and intensity of a high-resolution camera image for denser 3D reconstruction.

The second technique, Assisted stereo matching, aims at overcoming the weaknesses of stereo matching algorithms, by providing prior knowledge obtained from other depth sensors. One of the first proposed methods, Romero et al. [13], attempts to fuse the range information obtained from a laser range finder into the disparity computations of a trinocular stereo vision system. Guomundsson, Aanaes, and Larsen [14] also try to incorporate the depth data from a low-res Time-of-Flight camera into the stereo-matching problem. The disparity estimates, even though sparse in the stereo image plane, are used in the lowest level of a hierarchical Dynamic Programming algorithm. This way, the algorithm has an initial per pixel constraint on the disparity search space instead of having just an overall minimum-maximum disparity range initialization as in the standard case. On the other hand, Fischer, Arbeiter, and Verl [15] attempted to adapt the Semiglobal optimization approach (Hirschmuller [16]) to combine the data from a Time-of-Flight camera into stereo disparity computations. Finally, Huber and Kanade [17] proposed two ways to integrate LiDAR data in stereo matching: disparity space reduction and path promotion. The disparity space reduction approach aims to predict the expected disparity range interval for each pixel with LiDAR data's aid. Meanwhile, in Path promotion approach, a normal Dynamic Programming energy function is expanded to include a variational term, penalizing the deviation of disparity solution from the expected value obtained from LiDAR data.

Finally, with the growing interest in deep-learning, several recent works addressed the problem of utilizing the potential of Convolutional Neural Networks (CNN). The major drawback of such approaches is the need for huge training data set, adequately labeled with ground truth to train the network. Riegler et al. [18] introduce a machine learning approach to create a high-resolution depth image from a sparse, low-resolution depth map, and a registered image. The authors propose a fully-convolutional network (FCN) architecture, which is trained to create a high-res depth estimate from the input set, where the training data is generated from a simulation. Meanwhile, the Multi-Scale Guided convolutional network (MSG-Net) introduced by Hui, Loy, and Tang [19] upsamples the depth image in several progressive levels with multi-scale guidance from high-res intensity image. The intensity image is downsampled to the resolution of the input depth map, and feature extraction is performed at each level. The extracted depth feature map and image feature map are fused and fed to a deconvolution layer for upsampled reconstruction to the next level until the final resolution is achieved.

III. LiDAR-camera calibration

In a multi-sensor architecture, like the experimental vehicle employed for the tests, it is fundamental to accurately determine the sensors' relative pose before implementing a sensor fusion algorithm. Once the roto-translation of one sensor from another is available, a transformation could be defined, which can transform a 3D point in LiDAR coordinates to a 2D point in camera coordinates. In the literature, there are different methods for camera-LiDAR calibration; in [20] authors introduce the use of a triangular calibration target of known size where the vertices of a triangle will act as virtual constraints for the relative pose estimation. Pereira et al. [21] utilize a spherical target placed at predefined grid points on the ground in successive scans. For our vehicle, we implemented a solution based on the approach Similar to Guindel et al. [22], where the authors used a planar target with four symmetrically carved circular holes.

The calibration pattern is placed in the common FOV of both the sensors, such that the pattern can be captured in entirety by the camera, and each of the holes has to be intersected by at least two laser beams. The pattern board is then identified in the point cloud published by LiDAR using a plane segmentation procedure based on *Random Sample Consensus (RANSAC)*. To accomplish the same in the stereo camera reference frame, an accurate point cloud is created through Semiglobal Stereo matching, on which plane segmentation is performed. Subsequently, a circle segmentation procedure is carried out, followed by adding the center points to the cloud. The final registration is performed utilizing Iterative Closest

Point (ICP) algorithm, where the identified center points act as points of correspondence.

IV. Depth estimation pipeline

The process flow of the patch-wise depth correction approach can be divided into three steps: projection of LiDAR points into the camera image frame, upsampling the projection image with the aid of stereo depth map (termed as Seeding), and finally achieving a dense depth map using a special Bilateral Filter.

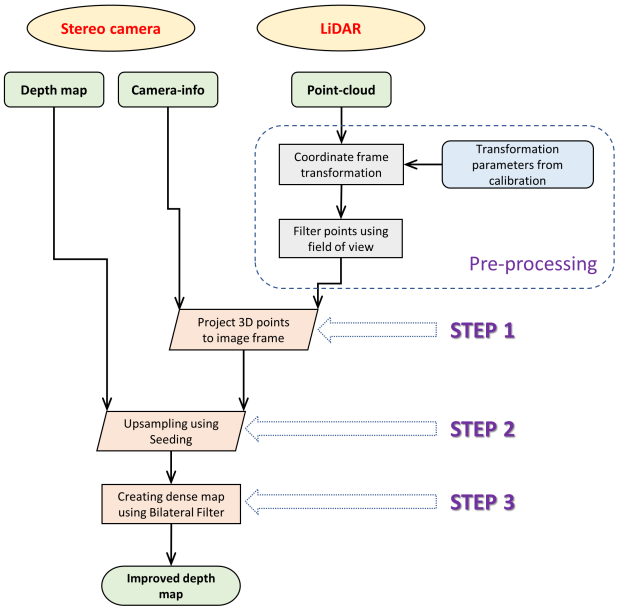


FIG. 1: Pipeline of the Depth Correction Algorithm

A. Projection of cloud points to the image frame

The first step of the depth correction algorithm is to project the 3D point obtained from LiDAR into the left camera frame of the stereo system. For this purpose, the LiDAR point cloud is transformed into the camera frame, using the transformation parameters yielded from the calibration. Subsequently, the points are filtered based on the camera FOV. Also, the points occluded in the camera line of vision due to perspective change are removed. For this, the following angular relationship of adjacent points in a laser ring is utilized,

$$\tan\left(\frac{x_j}{z_j}\right) > \tan\left(\frac{x_i}{z_i}\right) \quad \forall \; j > i \;\text{ counting from left} \tag{1}$$

Then for each 3D point $\mathbf{X}$, the pixel location in computer coordinates $\mathbf{x}_{R,l} = (u, v)$ is calculated using the following equation in homogeneous form:

$$\mathbf{x}_{R,l} = \mathbf{P}_l \mathbf{X} \tag{2}$$

where $\mathbf{P}_l$ is the projection matrix of the left camera. The depth map is created by encoding the depth value, which is the Z *coordinate* of 3D point, into the located pixel.

B. Upsampling using Seeding

The projection stage generally provides a very sparse depth image, depending on the resolution of the LiDAR used. This can be upsampled by merging information from the stereo camera depth map. A patch-wise correction approach is followed. Comparing the value in a pixel of the projection image $(d^l_{x,y})$ where depth information is available, to the corresponding pixel of stereo camera depth map $(d^c_{x,y})$, an offset is devised which can be extended to the neighboring pixels $(\{P\}_{x,y})$. This may not hold true as we move further away from this pixel, as the chances of variation in disparity calculation increases. Then the projection image can be upsampled around this pixel as follows.

$$offset = d^l_{x,y} - d^c_{x,y} \tag{3}$$

$$d^l_{x_i,y_i} = d^c_{x_i,y_i} + offset \quad \text{where} \quad (xi, yi) \in \{P\}_{x,y} \tag{4}$$

Portions of the depth image, which are not in the LiDAR field of view, rely on the depth map from the stereo camera alone. To ensure a smooth transition, the *offset* of the nearest projected point can be used. Another aspect to consider is the truncation applied to depth values by stereo matching algorithms. In such regions, the LiDAR measurement is propagated to neighboring pixels.

C. Creation of dense depth map

Seeding is significant in incorporating the information from the depth map of commercially available stereo cameras, where in-built SDKs handle disparity calculations. It is a practical, low-cost technique to upsample the LiDAR projection image and of supreme importance for low-resolution LiDARs. To create the final depth map, an efficient method must be introduced to estimate the range values in unsampled pixels with reasonable accuracy.

1) Local Interpolation: Such spatial interpolations are generally performed using a sliding window or mask of polygonal shape, enclosing the point of interest and estimating the desired point-value by *local interpolation*. A typical sliding window will be square in shape, even though some researchers like Lertrattanapanich and Bose [23] also explore solutions based on Delaunay-triangles. Regardless of the shape, the sliding window has to be chosen, guaranteeing a minimum number of sampled points inside to ensure the estimator's statistical significance, consistency, and efficiency.

The local interpolation has to be performed while preserving edges and object boundaries. Several edge-preserving filters are used in computer vision, like the Anisotropic Diffusion filter, proposed by Perona and Malik [24], and the Guided Filter introduced by He et al. [25]. Recently Premebida et al. [26] introduced a modified Bilateral Filter for depth map upsampling, which is the method employed in this project.

The sliding window $\mathbf{R}$ chosen is a square of size $m_r \times m_r$ in pixel units. The *bilateral filter* will estimate the depth value for the pixel at the center of the window $\mathbf{x}_0$ by interpolating inside the local region $\mathbf{R}$. Interpolation applies to both sampled and unsampled locations of the depth map. Let $\overline{r}_0$ be the range

distance r at $\mathbf{x}_0$ which is to be estimated and $\mathbf{p}_i = (\mathbf{x}_i, r_i), i = 1, 2, ...n$ be the set of sampled points in $\mathbf{R}$, then the bilateral filter can be formulated as

$$\overline{r}_0 = \frac{1}{W} \sum_{\mathbf{x}_i \in \mathbf{R}} G_{\sigma_s}(\mathbf{x}_i) G_{\sigma_r}(r_i) r_i \tag{5}$$

where G_{σ_s} weights the points $\mathbf{x}_i$ in terms of their relative position to $\mathbf{x}_0$, G_{σ_r} incorporates the influence of their range values in estimation and W is a normalization factor for the weights. We set G_{σ_s} to be inversely proportional to the Euclidean distance of a sampled point from the center of the mask, giving

$$G_{\sigma_s} = \frac{1}{1 + (\parallel \mathbf{x}_0 - \mathbf{x}_i \parallel)} \tag{6}$$

G_{σ_s} ensures smoothness within $\mathbf{R}$ by giving higher weightage to nearest neighbors, but it doesn't have any significant influence in controlling jump discontinuities. This sudden range differences from foreground to background objects can be taken care of by the choice of G_{σ_r}. Hence the weight term on range is chosen as

$$G_{\sigma_r} = \frac{1}{1 + (|r_0 - r_i|)} \tag{7}$$

where r_0 is the range value at the location $\mathbf{x}_0$. As mentioned earlier, only a few pixels in the depth map are currently sampled. Consequently, a convenient substitute to be provided when $\mathbf{R}$ is centered at an unsampled location. To that end, $r_0 = min(r_i), \forall r_i \in \mathbf{R}$ has been adopted for any unsampled $\mathbf{x}_0$.

2) Edge Preservation: Edges, indicating a foreground and background object in the field of view, will be characterized by a discontinuity in range values of the sampled points. Applying the bilateral filter as formulated by Equation 5 over the entire depth map, without analyzing the presence of such a discontinuity within the sliding mask $\mathbf{R}$, will result in loss of that edge information. This is tackled by performing clustering on the sampled points of $\mathbf{R}$ based on the range values, and the presence of an edge is detected if the number of clusters is more than one, $nc > 1$. The implemented clustering algorithm is inspired by Density-Based Spatial Clustering of Applications with Noise (DBSCAN), introduced by Sander et al. [27], which depends on the definition of a *distance function (DF)* and the parameter ϵ. The distance function is defined as

$$DF = \left| \frac{r_k - r_{k+1}}{r_k + r_{k+1}} \right|, \qquad k = 1, 2, ...n - 1. \tag{8}$$

where n is the number of sampled points in $\mathbf{R}$. $DF > \epsilon$ indicates the presence of a range discontinuity, and multiple clusters will be formed. The sampled points are sorted in terms of range for the easiness of clustering, and the value of ϵ is chosen close to $\epsilon \approx 0.1$. In the event of having $nc > 2$, i.e., more than 2 clusters are present in a region, a decision must be made to choose the clusters of interest. The cluster with the smallest average range value is chosen as cluster s_1, regardless of the number of remaining clusters. Cluster s_2 is selected based on the following rule if $nc > 2$: s_2 is chosen as the cluster with more points and, in case of clusters with the same number of points, s_2 corresponds to the one with the smallest average range other than s_1. If the sliding window $\mathbf{R}$ belongs to a region without a discrete edge, all the points will be sampled into one cluster, and Equation 5 will be applied to all the points in $\mathbf{R}$. When $nc = 2$, it is limited to only one of the clusters based on the value of a parameter $\lambda = np_1/np_2$, where np_1 and np_2 are the number of points belonging to the clusters s_1 and s_2. A threshold Thr is set such that when $\lambda \geq Thr$ the bilateral filter will run on points belong to s_1, else it runs on the points belong to s_2.

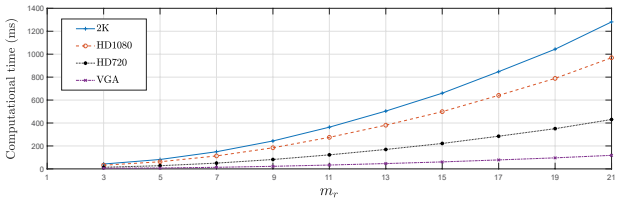


FIG. 2: Performance comparison for different resolution

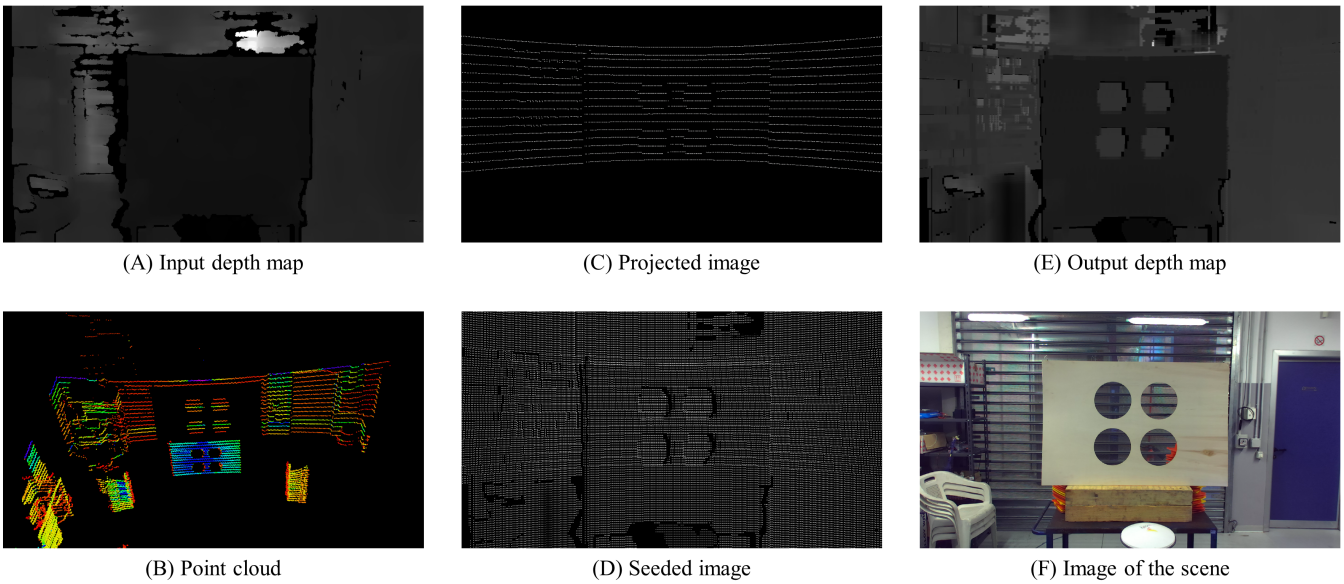
V. EXPERIMENTAL RESULTS

The proposed algorithm has been implemented on an instrumented vehicle. The processing pipeline of one of the scenes tested is explained in Figure 3. To ensure real-time performance, the algorithm runs on GPU. The resolution of the depth image and the sliding window size m_r of the local interpolation are found to be influencing the computational time, as seen in the plot of Figure 2.

A. Validation of results using KITTI data set

To evaluate the algorithm's accuracy, tests have been conducted on different scenarios collected from *the KITTI Vision Benchmark Suite*. Due to the algorithm's task-specific nature, a benchmarking comparison with prior solutions is not possible. Instead, the accuracy of the solution is compared against the original depth image from the stereo camera. Point cloud, left-right images, and the calibration parameters are gathered from *KITTI Raw Data* [6]. The corresponding ground-truth images are acquired from *KITTI Depth Evaluation data set* [28]. The ground-truth for depth evaluation was created using a process that incorporates a set of consecutive scans from the LiDAR (five before and five after the actual frame of interest), where this sequences of point clouds were conveniently merged by an ICP technique as reported in [6], followed by a manual correction step to rectify eventual ambiguities. The point cloud collected is downsampled to 16 channels to match the hardware specifications, and also input depth images are created from stereo images using semi-global stereo matching.

1) Role of sliding window size: The local interpolation described in Section IV-C1 is performed using a square mask $\mathbf{R}$ of size $m_r \times m_r$ in pixel units. Only the sampled points inside $\mathbf{R}$ will participate in the interpolation algorithm, which estimates depth value for the pixel at the center of the mask.



(A) Input depth map (C) Projected image (E) Output depth map

(B) Point cloud (D) Seeded image (F) Image of the scene

FIG. 3: Different stages of the depth processing pipeline. Input data are depth maps from the stereo-camera (A) and point cloud from the LiDAR (B). The point cloud is projected to the image frame to form the projected image (C), and later upsampled using the input depth map to form the seeded image (D). The final output is obtained by local interpolation (E). The image of the scene is also provided for reference (F).

Hence, the mask's size should be chosen, guaranteeing a minimum number of sampled points inside to ensure the statistical significance, consistency, and efficiency of the estimator.

If the window chosen size is too small, a large percentage of pixels in the image will be left unsampled. On the other hand, if the size is bigger than necessary, the algorithm will be computationally heavier to use in real-time applications. Also, using a bigger mask will result in the output depth values to be extensively smoothened, hence reduces the accuracy of estimation. This is particularly true for smaller or far away objects, which occupy only a few pixels in the image. The choice of m_r will also depend on the resolution of images. Low-resolution images generally require a small mask, while higher resolution images will demand a bigger mask for efficiency. A mask size of $m_r = 11$ to 13 was found to be most efficient on the KITTI images of resolution 1242×375, considering the accuracy of depth estimation, the density of depth map created, and the required computational time.

2) Role of seeding neighborhood: The definition of seeding neighborhood $\{P\}_{x,y}$ (explained in Section IV-B) also plays a vital role in the fusion of LiDAR and stereo data. During the experiments conducted on the testing hardware as well as the KITTI data set, $\{P\}_{x,y}$ is chosen to be a vertical stripe centered at the LiDAR sampled point. This was owing to the finer resolution of LiDARs in the horizontal direction. This will have to be modified for a different laser scanner, as per the horizontal and vertical sparsity of the projected image. Also, the seeding neighborhood's density, which is the closeness of pixels considered for seeding, can be varied. This will inherently influence the choice of m_r also, as a denser definition of $\{P\}_{x,y}$ will provide more points inside $\mathbf{R}$ for local interpolation.

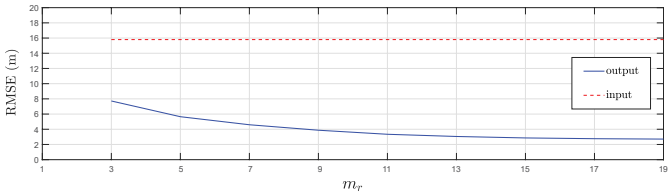


FIG. 4: $RMSE$ vs m_r

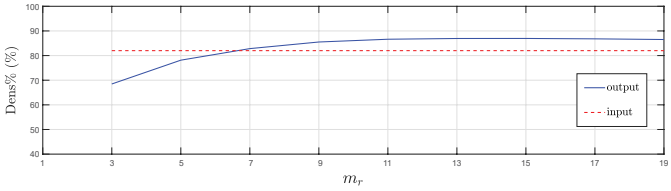


FIG. 5: $Dens\%$ vs m_r

3) Results and discussion: The results obtained from experiments on the KITTI data set proved the effectiveness of the depth correction algorithm. The output depth map is significantly superior to the input depth map, both in terms of density and depth reconstruction accuracy. This is demonstrated in Figure 6, where a reference image (A) and ground-truth depth map (B) are presented along with error maps of both input (C) and output (D) depth images. In the $RMSE$ vs m_r plot shown in Figure 4, root mean square errors (RMSE) in depth estimation at the sampled locations of ground-truth image are reported. The RMSE improves as m_r increases, and shows a saturation behaviour for $m_r > 13$. For very big window

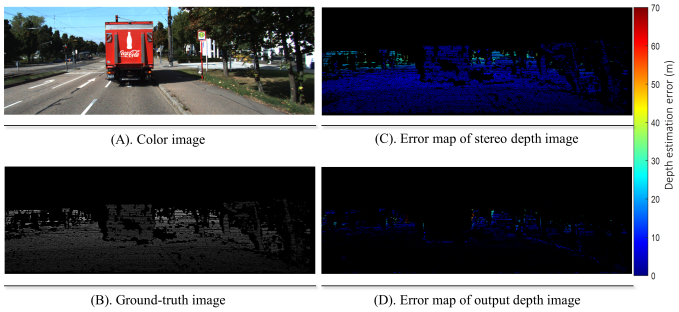


FIG. 6: Depth accuracy evaluation on an image from the Kitti dataset.

sizes ($m_r \geq 17$), the accuracy of estimation decreases in some cases. This is due to the over smoothening phenomenon generally observed in image filtering applications, where points significantly far from the point of interest influence the depth estimation.

In the plot $Dens\%$ vs m_r shown in Figure 5, the density of sampled points in image frame is reported. The output depth map is denser than the input stereo depth map, for mask sizes $m_r > 7$.

VI. CONCLUSIONS AND FUTURE WORKS

In this paper, we present a depth correction algorithm that produces an accurate depth map by fusing information from a stereo camera depth image and a LiDAR point cloud. Unlike the existing data fusion techniques for stereo cameras and LiDAR, the proposed method employs a post-processing strategy with redundant sensor configuration. The patch-wise correction approach rectifies stereo matching errors, especially in correcting errors caused by inadequate ambient illumination and texture-less surfaces. The algorithm's accuracy has been demonstrated by conducting tests on multiple scenarios from the KITTI data set. The implementation of the solution has been done by leveraging the power of modern GPUs and Compute Unified Device Architecture (CUDA) computation to allow real-time processing.

Future works will focus on merging information from color images to improve the overall reconstruction further.

REFERENCES

[1] P. Y. Shinzato, D. F. Wolf, and C. Stiller, "Road terrain detection: Avoiding common obstacle detection assumptions using sensor fusion," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*. IEEE, 2014.

[2] K. Kidono, T. Miyasaka, A. Watanabe, T. Naito, and J. Miura, "Pedestrian recognition using high-definition lidar," in *2011 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2011, pp. 405–410.

[3] F. Zhang, D. Clarke, and A. Knoll, "Vehicle detection based on lidar and camera fusion," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2014, pp. 1620–1625.

[4] A. Rangesh and M. M. Trivedi, "No blind spots: Full-surround multi-object tracking for autonomous vehicles using cameras and lidars," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 4, pp. 588–599, 2019.

[5] J. Prakash, M. Vignati, S. Arrigoni, M. Bersani, and S. Mentasti, "Tele-operated vehicle-perspective predictive display accounting for network time delays," in *ASME 2019 International Design Engineering Technical Conferences and Computers and Information in Engineering*. American Society of Mechanical Engineers Digital Collection, 2019.

[6] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[7] Q. Yang, R. Yang, J. Davis, and D. Nistér, "Spatial-depth super resolution for range images," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2007, pp. 1–8.

[8] D. Chan, H. Buisman, C. Theobalt, and S. Thrun, "A noise-aware filter for real-time depth upsampling," 2008.

[9] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele, "Joint bilateral upsampling," in *ACM Transactions on Graphics (ToG)*, vol. 26, no. 3. ACM, 2007, p. 96.

[10] J. Diebel and S. Thrun, "An application of markov random fields to range sensing," in *Advances in neural information processing systems*, 2006, pp. 291–298.

[11] S. Gould, P. Baumstarck, M. Quigley, A. Y. Ng, and D. Koller, "Integrating visual and range data for robotic object detection," 2008.

[12] R. Reulke, "Combination of distance data with high resolution images," in *ISPRS Commission V Symposium Image Engineering and Vision Metrology*, vol. 2, 2006, pp. 25–27.

[13] L. Romero, A. Núñez, S. Bravo, and L. E. Gamboa, "Fusing a laser range finder and a stereo vision system to detect obstacles in 3d," in *Ibero-American Conference on Artificial Intelligence*. Springer, 2004.

[14] S. Á. Guomundsson, H. Aanaes, and R. Larsen, "Fusion of stereo vision and time-of-flight imaging for improved 3d estimation," *International Journal on Intelligent Systems Technologies and Applications (IJISTA)*, vol. 5, no. 3/4, pp. 425–433, 2008.

[15] J. Fischer, G. Arbeiter, and A. Verl, "Combination of time-of-flight depth and stereo using semiglobal optimization," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 3548–3553.

[16] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 30, no. 2, pp. 328–341, 2007.

[17] D. Huber, T. Kanade *et al.*, "Integrating lidar into stereo for fast and improved disparity computation," in *2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*. IEEE, 2011, pp. 405–412.

[18] G. Riegler, D. Ferstl, M. Rüther, and H. Bischof, "A deep primal-dual network for guided depth super-resolution," *arXiv preprint arXiv:1607.08569*, 2016.

[19] T.-W. Hui, C. C. Loy, and X. Tang, "Depth map super-resolution by deep multi-scale guidance," in *European conference on computer vision*. Springer, 2016, pp. 353–369.

[20] S. Debattisti, L. Mazzei, and M. Panciroli, "Automated extrinsic laser and camera inter-calibration using triangular targets," in *2013 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2013, pp. 696–701.

[21] M. Pereira, D. Silva, V. Santos, and P. Dias, "Self calibration of multiple lidars and cameras on autonomous vehicles," *Robotics and Autonomous Systems*, vol. 83, pp. 326–337, 2016.

[22] C. Guindel, J. Beltrán, D. Martín, and F. García, "Automatic extrinsic calibration for lidar-stereo vehicle sensor setups," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017, pp. 1–6.

[23] S. Lertrattanapanich and N. K. Bose, "High resolution image formation from low resolution frames using delaunay triangulation," *IEEE Transactions on Image Processing*, vol. 11, no. 12, pp. 1427–1441, 2002.

[24] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 12, no. 7, pp. 629–639, 1990.

[25] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 6, pp. 1397–1409, 2012.

[26] C. Premebida, L. Garrote, A. Asvadi, A. P. Ribeiro, and U. Nunes, "High-resolution lidar-based depth mapping using bilateral filter," in *2016 IEEE 19th international conference on intelligent transportation systems (ITSC)*. IEEE, 2016, pp. 2469–2474.

[27] J. Sander, M. Ester, H.-P. Kriegel, and X. Xu, "Density-based clustering in spatial databases: The algorithm gdbscan and its applications," *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 169–194, 1998.

[28] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, "Sparsity invariant cnns," in *International Conference on 3D Vision (3DV)*, 2017.