2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

Real Time Dense Depth Estimation by Fusing Stereo with Sparse Depth Measurements

Shreyas S. Shivakumar, Kartik Mohta, Bernd Pfrommer, Vijay Kumar and Camillo J. Taylor

***Abstract*— We present an approach to depth estimation that fuses information from a stereo pair with sparse range measurements derived from a LIDAR sensor or a range camera. The goal of this work is to exploit the complementary strengths of the two sensor modalities, the accurate but sparse range measurements and the ambiguous but dense stereo information. These two sources are effectively and efficiently fused by combining ideas from anisotropic diffusion and semi-global matching.**

We evaluate our approach on the KITTI 2015 and Middlebury 2014 datasets, using randomly sampled ground truth range measurements as our sparse depth input. We achieve significant performance improvements with a small fraction of range measurements on both datasets. We also provide qualitative results from our platform using the PMDTec Monstar sensor. Our entire pipeline runs on an NVIDIA TX-2 platform at 5Hz on 1280×1024 stereo images with 128 disparity levels.

I. INTRODUCTION

Accurate real-time dense depth estimation is a challenging task for mobile robots. Most often, a combination of sensors is used to improve performance. Sensor fusion is the broad category of combining various sensors to produce better measurement estimates. These sensors are combined to compliment each other and overcome individual shortcomings. We focus on the fusion of high resolution image data with low resolution depth measurements, which is a common method of obtaining dense 3D information.

Passive stereo cameras are a popular choice for 3D perception in mobile robots, able to generate dense depth estimates that are readily scaled by increasing the resolution of the sensors used. However, stereo depth estimation algorithms are typically dependent upon visual cues and scene texture and can struggle to assign disparities in regions that contain of uniform patches, blurred regions and large illumination changes. Depending on the resolution and performance desired, dense stereo based depth estimation can be computationally demanding on compute constrained robot platforms. However, embedded hardware accelerators such as the Nvidia TX-2 can exploit the parallelism inherent in the stereo matching algorithms making these approaches practical for robotic applications [1].

LIDAR sensors are a popular choice for accurate and efficient depth estimation. These sensors are expensive and are often heavy to mount on a small robot platforms such as an unmanned aerial vehicle. Time-of-flight devices such as the PMDTec Monstar [2] provide depth estimates at a low resolution. They provide accurate short range depth measurements and are often used in indoor robotics and low light environments. Unlike photogrammetry based stereo, this sensor performs very well on surfaces with uniform appearance such as flat plain walls. Phase difference between the emitted and returned infrared signals are used to measure distances, and sensors such as the PMD are a practical alternative that we have successfully used on our unmanned aerial vehicle platform for indoor obstacle avoidance and mapping. (Figure 1)

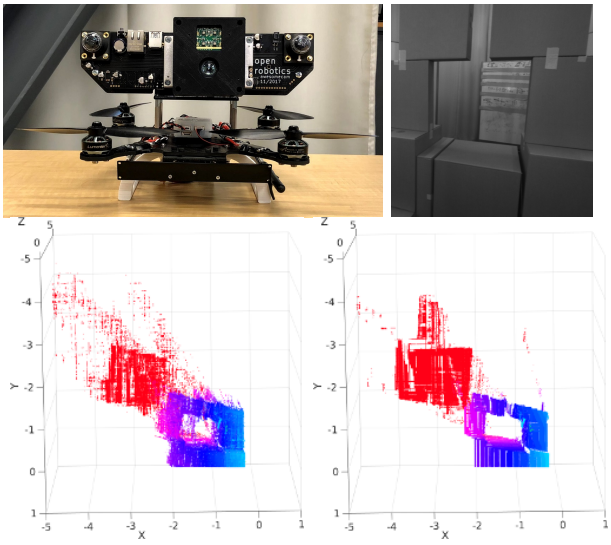

Fig. 1. a) Top-L: Our stereo camera with a PMD Monstar placed roughly in the middle for optimal overlap with the stereo imagers, mounted on our Falcon 250 UAV platform. b) Top-R: A grayscale image collected from our stereo setup c) Bottom-L: Corresponding point cloud generated using Semi Global Matching d) Bottom-R: Corresponding point cloud generated using our Diffusion based approach - by fusing the two sensors, we can obtain high resolution depth estimates that are robust to the noisy measurements that is often seen in regular stereo based depth estimation. Colors are mapped between 0 - 2.5 meters. The red surface represents the curtain viewed through the aperture in the cartons (in blue)

Beder et al. compare both sensing approaches under optimal conditions and concluded that the PMD performed with better accuracy in surface reconstruction, but that an ideal setup would be a fusion of both systems to overcome the low resolution of the PMD [3]. Scharstein et al. discuss several different approaches to stereo depth estimation [4]; though slightly dated, this work still presents a comprehensive list. Modern approaches use Convolutional Neural Networks to solve the disparity estimation and patch matching problem,

We gratefully acknowledge the support of DARPA grants HR001151626 and HR0011516850, ARO grant W911NF-13-1-0350, ONR grant N00014-07-1-0829 and USDA grant 2015-67021-23857

Shreyas S. Shivakumar, Kartik Mohta, Bernd Pfrommer, Vijay Kumar and Camillo J. Taylor are with the GRASP Laboratory, School of Engineering and Applied Sciences, University of Pennsylvania, Philadelphia PA 19104 {sshreyas,kmohta,bpfrommer,kumar,cjtaylor}@seas.upenn.edu

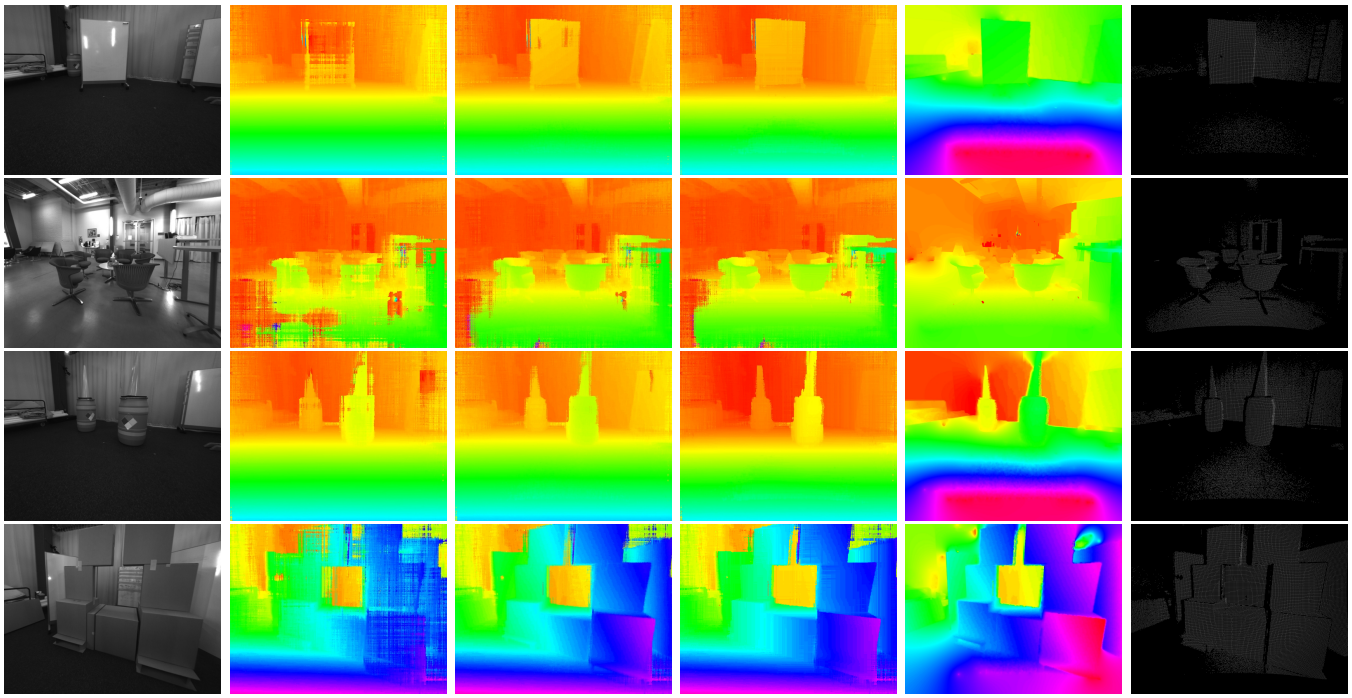

Fig. 2. (PMD Monstar Dataset) L-R: Rectified left image (grayscale); Results of the Semi-Global Matching - the algorithm performs poorly on uniform textures such as the whiteboard and the cardboard cartons, common to most stereo algorithms; Neighborhood Support method - performs slightly better than SGM but still has trouble filling in erroneous stereo estimates; Diffusion based method - performs significantly better than both the above methods, while preserving disparity discontinuities and retaining accurate disparity measurements; Anisotropic Diffusion - performs well at regions where PMD measurements exist, but greatly misrepresents disparity measurements at regions far away from such points, as one could expect from a monocular only setup; PMD Monstar points used after filtering out noisy measurements

however these networks are often too computationally demanding to run in real-time at our desired resolution [5]. We build our method upon the seminal work of Hirschmuller et al. on Semi Global Matching (SGM) stereo algorithm, a method widely popular even today. [6].

Huber et al. present a lidar and stereo integration [7] method that reduces the disparity search space and reduces computation time by a factor of five but assumes the lidar measurements are almost uniformly distributed over the image. Their second method, a dynamic programming based approach, uses the lidar points in the optimization process, however no quantitative results are provided for comparison with ours. Maddern et al. propose a probabilistic method of fusing sparse 3D lidar data with dense stereo data by modeling the error characteristics of both sensors and deploy it on a low-power GPU system [8]. They use lidar measurements as support points for the stereo method mentioned in [9]. Veitch-Michaelis et al. also propose a similar approach with a region growing based stereo algorithm [10].

A survey of ToF-stereo fusion is presented by Nair et al. [11] and we describe some of these solutions. Kuhnert et al. introduce an early variant of the PMD sensor and present a direct approach of fusing the PMD depth estimates with a WTA-style stereo algorithm [12]. Kahne et al. present a PMD and dense stereo fusion pipeline, where PMD measurements are used to reduce the disparity search space and the disparity estimation is set up as a graph cuts problem [13]. This work does not present any quantitative estimates of accuracy improvements or computational feasibility for real time systems. Zhu et al. present the ToF-stereo fusion problem as a Belief Propagation problem using Markov Random Fields, where the weights correspond to confidence in each sensor's measurements [14]. The work of Gandhi et al. [15] presents a similar ToF-stereo fusion pipeline, where the ToF depth points are seeds in a stereo seed growing algorithm. Their pipeline is validated on their own dataset as well as the Middlebury dataset, where points are uniformly sampled from the ground truth. Gudmundsson et al. use a ToF camera and use the range values to constrain a hierarchical stereo matching algorithm [16].

A different, but relevant method involves using the range measurements along with a monocular image to generate dense depth by guided interpolation. Courtois et al. use bilateral filtering based interpolation of lidar data for the purpose of robot mapping [17]. This paper presents quantitative results on the KITTI dataset, and compares their method to previous methods. Prembida et al. propose an interpolation and up-sampling based method of obtaining dense disparity estimates from LIDAR scans [18]. Ma et al. propose a deep learning approach to depth up-sampling and are currently in the top 3 on the KITTI depth completion benchmark [19] [20].

Fischer et al. present work that is similar to ours, where ToF depth information is filtered and used to guide the cost aggregation stage of the Semi-Global Matching algorithm [21]. The proposed method uses ToF depth to limit the search space during pixel-wise matching if the ToF data is in approximate agreement of the naïve matching estimate. This

approach is intuitively similar to our neighborhood support mentioned in Section II.

The main **contributions** of our work are: *(1)* a method of integrating sparse accurate depth measurements into a dense stereo depth estimation framework, combining traditional stereo range fusion and depth interpolation techniques; *(2)* a quantitative evaluation of our method on KITTI 2015 and Middlebury 2014 datasets and a small computational footprint allowing real time dense depth estimation on computationally constrained mobile robots. We have also made our code publicly available[1].

II. TECHNICAL APPROACH

A. Pipeline

Our processing pipeline obtains as input a pair of rectified stereo images I (left camera) and J (right camera). Additionally, using the calibrated intrinsics and extrinsics, we convert the depth sensor's range measurements into a depth image in the left camera's reference frame with matching focal length.

Setting up the cost volume: As is common in many stereo algorithms, we first transform our grayscale intensity image to a feature space more robust to intensity variations. We apply the census transform to a window around each pixel in the left and right image and the resulting bitvectors are denoted by $I_{\text{cen}}(x, y)$ and $J_{\text{cen}}(x, y)$ [22]. A 3D cost volume is then computed, where the X and Y axes correspond to the 2D image co-ordinates and the Z axis to the disparity range. Each element of this volume $C((x, y), d)$ represents a cost, or similarity between the transformed value in the left image and its corresponding value in the right, displaced in the y-axis by d, where $d = 1..D_{MAX}$ as seen in Eq 1,

$$C((x, y), d) = SIM(I_{\text{cen}}(x, y), J_{\text{cen}}(x, y - d)) \quad (1)$$

Here, the similarity measure $SIM(a, b)$ is the Hamming distance between the two census bit vectors from the left and right images.

Cost aggregation: We follow the aggregation method proposed by Hirschmuller et al. [6]. The aggregation step is formulated as an energy minimization equation, reminiscent of scanline optimization based stereo methods, done along multiple different directions at every pixel. To reduce the computational complexity, we consider 4-8 directions, instead of the original 16. The SGM algorithm proceeds by computing aggregate costs along a number of different directions as described in Eq 2.

$$C'_r(p, d) = C(p, d) + min \begin{cases} C'_r(p-1, d) \\ C'_r(p-1, d+1) + P_1 \\ C'_r(p-1, d-1) + P_1 \\ min_i C'_r(p-1, i) + P_2 \end{cases} \quad (2)$$

Where C'_r is the aggregated cost volume for a given path r and p indicates a point along a path r. The notation $p-1$ indicates the point previous to p along the direction r. The penalty terms P_1 and P_2 indicate how heavily to penalize small disparity differences and large ones respectively. The final disparities are calculated by summing over the different paths r and selecting the disparity level d with the lowest cost.

$$D(x, y) = \arg\min_d S((x, y), d) \quad (3)$$

$$S((x, y), d) = \sum_r C'_r((x, y), d) \quad (4)$$

We keep this energy minimization formulation as is, and seek to introduce the depth measurements during the cost volume construction phase.

Updating the cost volume: We introduce our range measurements at the cost volume creation stage, making updates to element $((x, y), z)$ in the cost volume at points where there is a measured disparity value. We denote these elements by $((x_m, y_m), d_m)$. These measured readings are treated as high confidence disparity estimates and are used to modify the original cost volume entries. We propose three different approaches,

1) Naïve Fusion: The naïve approach involves setting the cost at the measured point (x_m, y_m, d_m) to be a very small value, 0. Intuitively, for a given pixel this means that we have the highest confidence that the disparity to be assigned to this point is d_m.

$$C((x_m, y_m), d_i) = \begin{cases} 0 \text{ if } d_i = d_m \\ C((x_m, y_m), d_i) \text{ otherwise} \end{cases} \quad (5)$$

By naïvely altering the cost elements $((x_m, y_m), d_m)$, we see an improvement over the basic SGM algorithm, however, the nature of the cost aggregation makes it robust to points that are in strong disagreement with their neighbors, both spatially and along the disparity axis. Therefore intuitively, the aggregation procedure tries to reject or ignore very sparse updates made to the cost volume that are in strong disagreement with the original stereo costs. Additionally, the spread of information is limited to the paths along which an update was made, and if the range measurements are too sparse, significant improvement is not observed.

2) Neighborhood Promotion: As a solution to the above problem, we propose the following method: Since we are confident about the range measurements that we obtain from our range sensor, we can force lowered costs or energies on points in the image that neighbor these sparse locations, essentially providing more guidance for the energy minimization. We use the grayscale image as the guide, assuming that within small windowed regions, the grayscale intensities of two points on a surface having similar depth also have similar intensities. For every range measurement $((x_m, y_m), d_m)$, we observe the grayscale intensities of its neighbors and calculate a set of weights based on the intensity difference between the point (x_m, y_m) and its neighbors, within a window of radius K_w. The weight matrix $W_m(i, j)$ is calculated using a Gaussian with a smoothing parameter σ_r.

$$W_m(i, j) = G_{\sigma_r}(I(i, j) - I(x_m, y_m)) \quad (6)$$

[1]https://github.com/ShreyasSkandanS/stereo_sparse_depth_fusion

Therefore for each window region, a cost update is made to all pixels in this region. For each disparity level k, the cost updates are as follows,

$$C((x_m, y_m), d_k) = \begin{cases} \beta \text{ if } |d_k - d_m| \geq \tau_d \\ \epsilon \text{ otherwise} \end{cases} \tag{7}$$

And the cost update to the neighboring pixels is,

$$C((x_i, y_j), d_m) = \begin{cases} (1 - W_m(i,j))\beta \text{ if } W_m(i,j) < \tau_n \\ \epsilon \text{ otherwise} \end{cases} \tag{8}$$

where (i, j) are co-ordinates for the points with respect to (x_m, y_m) within the window region of radius K_w. Notice that we now also introduce a parameter τ_d, which controls the degree of belief in our measured disparity accuracy. And to further propagate our belief in our measured disparity d_m, we set all other disparity costs to be some large constant β. Here ϵ is the minimum cost assigned. The cost update made along the Z-axis between k and τ_d can be modeled either by some prior noise model associated with the sensor or by assigning constant values or linear gradients. We observed that a good value for τ_d is 2 for the PMD measurements and that a smooth interpolation of costs between τ_d and k does not show substantial improvement versus a constant cost update for small values of τ_d. Similarly, the threshold τ_n determines how similar two intensity values must be in order to believe that they are part of the same surface.

3) Diffusion Based Update: With the intuition that larger support regions provide more information to the optimization procedure, we propose a third approach based on Anisotropic Diffusion and depth interpolation to update the volume [23]. Intuitively, we interpolate the sparse depth points from the range sensor and then use this information during the update step. We restrict this interpolation to regions near valid range measurements as points further away may not be part of the same surface. When using a PMD sensor, the points measured are usually close to the robot and trying to interpolate values far away leads to large inaccuracies in the resulting disparity.

Our interpolation limits are defined by a radius K_{interp} around each measured point. Points with valid measurements remain unaltered and the remaining points are assigned disparity values by leveraging the grayscale images for interpolation. Each interpolated disparity value is a weighted combination of all of the sparse disparity measurements within a radius of K_{interp} pixels of that location.

$$D(x, y) = \frac{\sum_{i,j} W(i,j) d(i,j)}{\sum_{i,j} W(i,j)} \tag{9}$$

where (i, j) are co-ordinates of measured disparity estimates from the sensor that are within K_{interp} pixels from (x, y). The weights $W(i, j)$ are calculated using a bilateral filtering method while $d(i, j)$ are disparity measurements from the range sensor [24]. Bilateral filters are commonly used edge preserving filters and in our case, depth discontinuity preserving. The pixel distance based smoothing is parameterized by σ_d.

$$\begin{aligned} W(i,j) = G_{\sigma_r}(I(i,j) - I(x_m, y_m)) \\ \times G_{\sigma_d}(|(i,j) - (x_m, y_m)|) \end{aligned} \tag{10}$$

For computational efficiency, we compute $\sum_{i,j} W(i,j)d(i,j)$ and $\sum_{i,j} W(i,j)$ separately and perform the division upon completion. Additionally, this can be computed in parallel over the set of points within the K_{interp} region around each measured point. Maintaining the sum of the weights also provides us with additional information regarding how similar a point at (x, y) is to its nearest measured point (x_m, y_m) which serves as a proxy for our confidence in the interpolated disparity value. The update step is then,

$$C((x_i, y_j), d_k) = \begin{cases} (1 - W(i,j))\gamma & \text{if } \tau_l < W(i,j) < \tau_u \\ \gamma & \text{if } W(i,j) \leq \tau_l \\ \epsilon & \text{if } W(i,j) \geq \tau_u \\ \beta & \text{if } |d_k - d_v| \geq \tau_d \text{ and } W(i,j) > \tau_l \end{cases} \tag{11}$$

Here, we use $d_v = D(x_i, y_j)$ to refer to interpolated depth sensor disparities at the point (x_i, y_j). The parameter τ_m controls our confidence in the interpolated value, using weight $W(i, j)$. Parameters ϵ, β and τ_d remain the same as before. Here, γ is some large penalty, similar or equal to β. The parameters τ_u and τ_l indicate a confidence range over the normalized weights W, representing a cutoff for high confidence and low confidence respectively.

III. RESULTS AND ANALYSIS

We discuss the performance of our algorithm on three datasets providing quantitative results on the KITTI 2015 and Middlebury 2014 datasets and qualitative results on our own PMD Dataset. On each dataset, we compare to the standard *Semi Global Matching* algorithm, without the left-right consistency check. For consistency we use the same P_1 and P_2 parameters for all images within a dataset, for all methods relying on SGM. Across datasets, we manually select P_1 and P_2 values after a parameter search. We choose D_{MAX} to be 256, which is sufficient for both our dataset as well as KITTI. On Middlebury 2014 there are some points with larger disparity values but we only evaluate points within this range.

We also compare against an *Anisotropic Diffusion* based approach to depth enhancement [25]. It must be noted that this method is independent of stereoscopic information and takes as its input a single image and a set of disparity points and generates a dense disparity image by diffusing these disparity points, using the input image as a guide.

1) KITTI 2015 Dataset: We evaluate these methods on the 200 stereo pairs provided. However, the ground truth data is not for every pixel in the grayscale image, but an accumulation of Lidar points over a range of frames before and after the reference image. For our evaluation, we use a subset of these measurements as our sparse depth input along with the stereo pair and our evaluation is done on ground truth measurements outside of this sample set. We

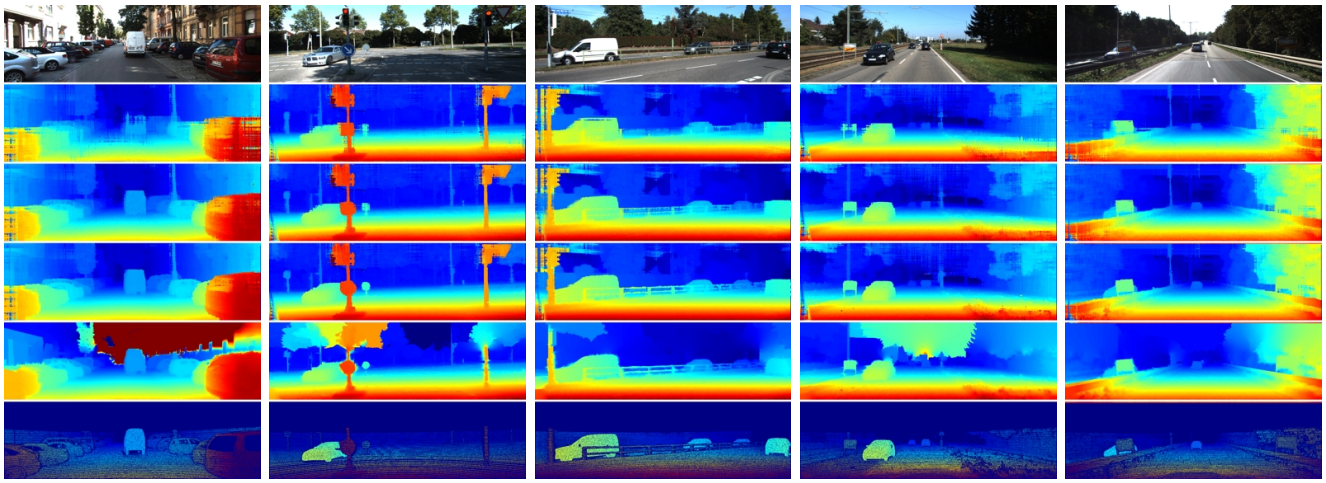

Fig. 3. (KITTI 2015) T-B: Rectified color image; Results of Semi Global Matching; Neighborhood Support method; Diffusion based method; Anisotropic Diffusion - note that the algorithm struggles to extrapolate a disparity value at regions where no seed range points exist and fails completely in certain cases; Ground truth points - these are the points at which the algorithm is evaluated, this is the original ground truth data with our sampled seed points removed. For these illustrations that is 15% of the total points available.

TABLE I
KITTI 2015 RESULTS

Each element in the table represents the percentage of pixels with disparity error greater than 1, 2 and 3 disparities away from ground truth.

Method	>1px	>2px	>3px
SGM	17.10	10.60	7.73
Naïve	15.77	9.88	7.22
Neighborhood Support	12.93	4.09	2.59
Diffusion Based	**4.26**	**2.01**	**1.51**
Anisotropic Diffusion	5.56	3.52	2.58

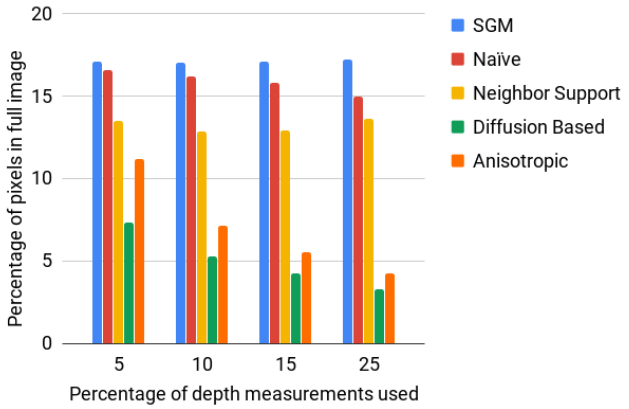

Fig. 4. Plot of error rates (>1px) versus number of samples for 5, 10, 15 and 25% of available ground truth lidar measurements. Evaluation is performed on the remaining samples.

randomly sample 15% of the ground truth depths for our final evaluation. We use the development kit provided with the dataset and report our errors from one to three disparity values, with a small tolerance as specified. The error rates are shown in Table I. A few example disparity maps are shown in Figure 3. Therefore, it is clear that our *Diffusion Based* method outperforms the others. Interestingly, *Anisotropic Diffusion* performs reasonably well. But this can be attributed to the fact that evaluation is performed on points where ground truth exists, and this is spatially in proximity to where our points are sampled from, even though randomly sampled. This algorithm however suffers from the interpolation flipping problem previously mentioned, as can be seen in Figure 3 a,b,d. The naïve approach does not significantly outperform *SGM*, and thus reaffirms our observation that creating larger areas of low energy around measured disparity levels in the cost volume is important for improving performance.

We also plot the error rates, which increase in the number of sampled points. This is shown in Figure 4. An interesting observation is the increase in error at 25% samples for the *Neighborhood Support* method. This is because for a fixed window size, an increase in the density of samples, will cause conflicting update to neighborhood regions, a problem that we solve in the *Diffusion Based* method by taking a weighted average of all support measurements. At the time of writing, the current state of the art on the KITTI 2015 Stereo benchmark achieves an error of 1.74% in the 3px error range. Though not directly comparable, we are able to achieve similar performance with our *Diffusion Based* method, scoring 1.51% on the training dataset provided, using only a small fraction of the lidar measurements.

2) Middlebury 2014 Dataset: For Middlebury 2014, we evaluate our algorithm on 23 of the stereo pairs provided, which have dense ground truth measurements. Since this dataset provides accurate, dense ground truth, we sample 2.5% of the total ground truth points, and randomly add noise to the measurements (up to 5%). These measurements are then used as before. As can be seen in Table II the *Diffusion Based* method achieves the lowest error rate, which can also be qualitatively seen in Figure 5. The anisotropic diffusion method also yields impressive results, but sees significant error due to misinterpreting depth boundaries on appearance alone. The *naïve* method doesn't show significant improvement over *SGM*, while the neighborhood support method performs notably better.

3) PMD Monstar Dataset: For our dataset, we use a pair of Python1300 CMOS sensors with 2.8mm, 1/2in sensors and FOV 95.3 x 82.6 degrees. The PMD has resolution of 352 x 287, and FOV of 100x85 degrees and is shown in Figure 1.

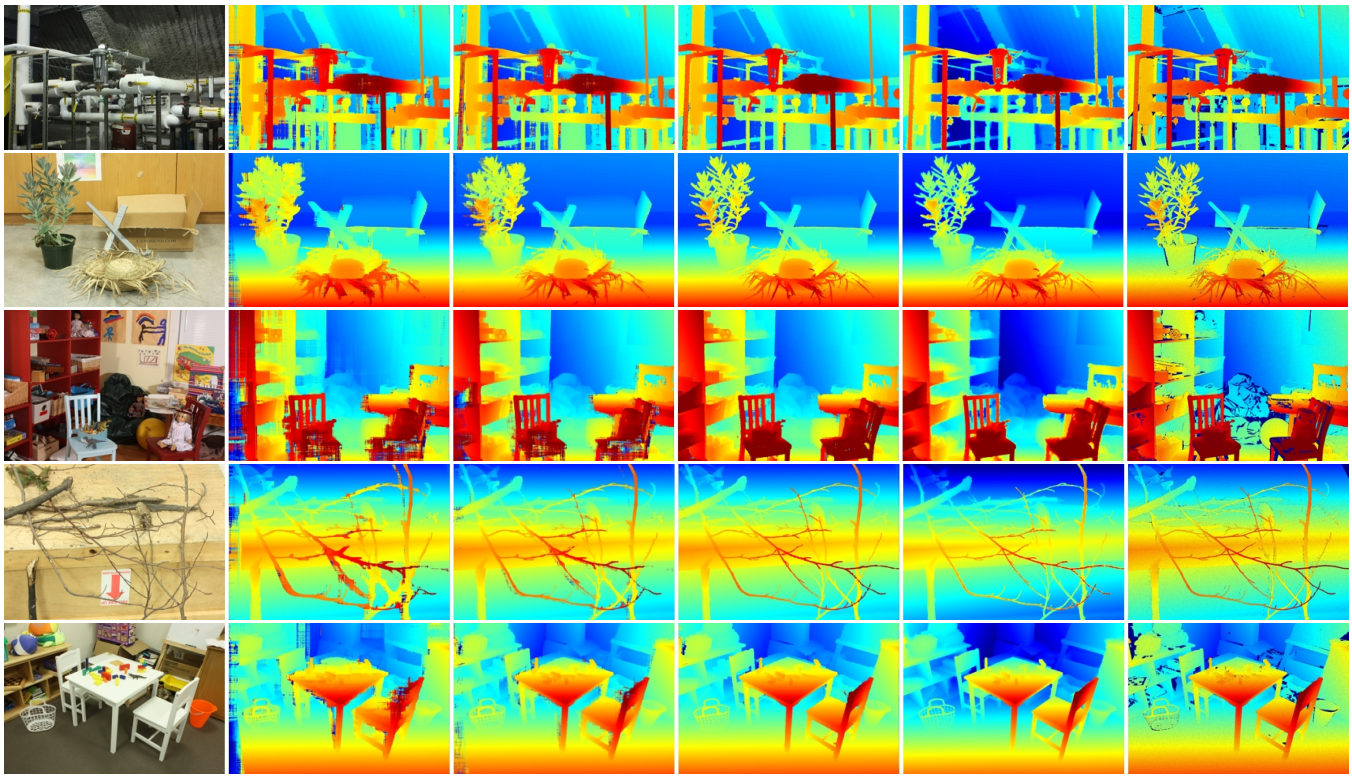

Fig. 5. (Middlebury 2014) L-R: Left rectified *image* (full resolution), *Semi-Global Matching* results - same P_1 and P_2 values were used for all images in this dataset and for all following methods. Here D_{MAX} was chosen to be 256; *Neighborhood Support* method - shows better performance when compared to SGM and our naïve approach (not pictured), but this algorithm still has problems with edges and partial occlusions; *Diffusion based* - this method performs the best, showing robustness to noise and preserving edges even with a small number of points; *Anisotropic Diffusion* - this method performs well at preserving edges and interpolating disparities, but regions where no nearby measurements exist result in incorrect disparities; *Ground Truth*.

TABLE II

MIDDLEBURY 2014 RESULTS

Each element in the table represents the percentage of pixels with disparity error greater than 1 disparity away from ground truth.

Method	**>1px**
SGM	3.4037
Naïve	3.2801
Neighborhood Support	1.8067
Diffusion Based	**0.1921**
Anisotropic Diffusion	0.4297

The stereo camera pair is calibrated with the PMD to obtain accurate intrinsic and extrinsic estimates for all sensors. The stereo baseline is 20cm, with the PMD placed in the center. Range measurements from the PMD are projected onto the stereo cameras and depth measurements are transformed to disparity estimates in the stereo domain.

Since we lack ground truth information to verify our claims, we qualitatively discuss performance on this data. We notice a consistent improvement in and around regions where PMD measurements exists. Depth discontinuities are more accurate and edges are well preserved in both the Diffusion Based method as well as the Neighborhood Support method. We show these results in Figure 2. Our *Diffusion Based* method is able to effectively use PMD measurements from surfaces such as the white-board in Figure 2a. Similar improvement is seen in the office setting image Figure 2b, where *SGM* struggles to assign correct disparities to the floor and chair regions. An extreme example is seen in Figure 2d where *SGM* incorrectly estimates disparities on the cartons and the white-board. *Anisotropic Diffusion* performs well, having only monocular and range information to work with, but suffers in regions where nearby range information doesn't exist. Since it is heavily influenced by image intensity gradients, edges translate to depth discontinuities, and this becomes a problem when no range information exists nearby, causing the interpolated disparity to ambiguously flip, introducing depth discontinuities where none exist.

IV. CONCLUSIONS

In summary, the present work explores several different means of incorporating sparse depth sensor measurements into a dense stereo algorithm. We evaluate different approaches on the KITTI and Middlebury 2014 datasets, and demonstrate how they improve upon an image-only based stereo vision approach. Naïvely incorporating the depth data holds only marginal advantage, but propagating depth data points to neighboring regions yields much improved results. The lowest error statistics, as well as good qualitative results, are obtained by combining a census based stereo cost function with an image edge preserving interpolation of the depth measurements, followed by the SGM procedure. Using the same approach in a monocular setting is demonstrated to suffer from serious artifacts, highlighting the importance of utilizing the stereo disparities. Future work includes modeling of sensor and calibration noise characteristics to adaptively select confidence thresholds.

REFERENCES

[1] D. Hernandez-Juarez, A. Chacón, A. Espinosa, D. Vázquez, J. C. Moure, and A. M. López, "Embedded real-time stereo estimation via semi-global matching on the gpu," *Procedia Computer Science*, vol. 80, pp. 143–153, 2016.

[2] H. Kraft, J. Frey, T. Moeller, M. Albrecht, M. Grothof, B. Schink, H. Hess, and B. Buxbaum, "3d-camera of high 3d-frame rate, depth-resolution and background light elimination based on improved pmd (photonic mixer device)-technologies," *OPTO, Nuernberg, May*, 2004.

[3] C. Beder, B. Bartczak, and R. Koch, "A comparison of pmd-cameras and stereo-vision for the task of surface reconstruction using patchlets," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007, pp. 1–8.

[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. 1-3, pp. 7–42, 2002.

[5] J. Zbontar and Y. LeCun, "Stereo matching by training a convolutional neural network to compare image patches," *Journal of Machine Learning Research*, vol. 17, pp. 1–32, 2016.

[6] H. Hirschmuller, "Accurate and efficient stereo processing by semi-global matching and mutual information," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2. IEEE, 2005, pp. 807–814.

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

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

[9] A. Geiger, M. Roser, and R. Urtasun, "Efficient large-scale stereo matching," in *Computer Vision–ACCV 2010*. Springer, 2010, pp. 25–38.

[10] J. Veitch-Michaelis, J. Muller, J. Storey, D. Walton, and M. Foster, "Data fusion of lidar into a region growing stereo algorithm," *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 40, no. 4, p. 107, 2015.

[11] R. Nair, K. Ruhl, F. Lenzen, S. Meister, H. Schäfer, C. S. Garbe, M. Eisemann, M. Magnor, and D. Kondermann, "A survey on time-of-flight stereo fusion," in *Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications*. Springer, 2013, pp. 105–127.

[12] K.-D. Kuhnert and M. Stommel, "Fusion of stereo-camera and pmd-camera data for real-time suited precise 3d environment reconstruction." in *IROS*, 2006, pp. 4780–4785.

[13] U. Hahne and M. Alexa, "Combining time-of-flight depth and stereo images without accurate extrinsic calibration," *International Journal of Intelligent Systems Technologies and Applications*, vol. 5, no. 3-4, pp. 325–333, 2008.

[14] J. Zhu, L. Wang, R. Yang, and J. Davis, "Fusion of time-of-flight depth and stereo for high accuracy depth maps," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[15] V. Gandhi, J. Čech, and R. Horaud, "High-resolution depth maps based on tof-stereo fusion," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4742–4749.

[16] S. A. Gudmundsson, H. Aanaes, and R. Larsen, "Fusion of stereo vision and time-of-flight imaging for improved 3d estimation," *International Journal of Intelligent Systems Technologies and Applications*, vol. 5, no. 3-4, pp. 425–433, 2008.

[17] H. Courtois and N. Aouf, "Fusion of stereo and lidar data for dense depth map computation," in *Research, Education and Development of Unmanned Aerial Systems (RED-UAS), 2017 Workshop on*. IEEE, 2017, pp. 186–191.

[18] C. Premebida, L. Garrote, A. Asvadi, A. P. Ribeiro, and U. Nunes, "High-resolution lidar-based depth mapping using bilateral filter," in *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*. IEEE, 2016, pp. 2469–2474.

[19] F. Ma and S. Karaman, "Sparse-to-dense: depth prediction from sparse depth samples and a single image," *arXiv preprint arXiv:1709.07492*, 2017.

[20] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, "Sparsity invariant cnns," in *2017 International Conference on 3D Vision (3DV)*. IEEE, 2017, pp. 11–20.

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

[22] R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," in *European conference on computer vision*. Springer, 1994, pp. 151–158.

[23] 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.

[24] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Computer Vision, 1998. Sixth International Conference on*. IEEE, 1998, pp. 839–846.

[25] J. Liu and X. Gong, "Guided depth enhancement via anisotropic diffusion," in *Pacific-Rim Conference on Multimedia*. Springer, 2013, pp. 408–417.